

LA DIGITALISATION VIDEO
UN EMULATEUR COMPLET DE L'IBM PC
2 SUPER PROGRAMMES EN BASIC

BELGIQUE : 220 FB - CANADA 6 $C - SUISSE : 10 FS

les plus de Micro Application pour Atari ST





PROGRAMMES SPOOLER RAM DISK HARDCOPY 149 F SEULEMENT

ATARI ST TRUCS et ASTUCES
Un recueil complet de trucs et d'astuces que votre tout nouveau ATARI ST va beaucoup apprécier ! Des graphismes fantastiques à partir de programmes en BASIC, des exemples et des conseils pour programmer en langage C et en Assembleur...
Réf. : ML 140
Prix : **149** FF

LE LIVRE DU GEM SUR ATARI ST
Cet ouvrage contient ce qu'il est nécessaire de savoir pour utiliser GEM efficacement : fonctionnement de la souris, Virtual Device Interface, Application Environment Services, Graphics Device Operating System. Description de routines utilisant GEM en C et en Assembleur. Utilisation standard et spécifique de l'operating system.
Réf. : ML 139
Prix : **149** FF

LE LIVRE DU LANGAGE MACHINE DE l'ATARI ST
Tout ce qu'il faut savoir pour tirer au mieux parti de votre ATARI ST : système de calcul et de bit manipulation du 68000, utilisation des registres, structure des commandes, programmation structurée : récursion, piles, procédures et fonctions, listings sources de programmes assembleurs, routines systèmes... Un super livre !
Réf. : ML141
Prix : **149 FF**

LA BIBLE DE L'ATARI ST
Ce livre contient un ensemble complet d'informations sur l'ATARI ST, la description HARDWARE de la machine ainsi que des schemas détaillés et amplement expliqués, notamment des interfaces V24, du port d'extension, de l'interface midi, la structure des graphiques, du BIOS, de GEM, les adresses systemes importantes, le fonctionnement de la souris...
Réf. : ML 142
Prix : **249** FF

PEEKS & POKES
Voulez-vous profiter rapidement de votre ATARi ST? Bien sûr, mais comment faire ? Eh bien lisez cet ouvrage et vous découvrirez vite tout ce que vous pouvez tirer facilement de votre ST. Les domaines couverts sont : la configuration hardware, la memoire, le TOS, GEM, les différents langages (LOGO, BASIC, C, Assembleur), programmation des graphismes et des sons, techniques de programmation, gestion des disquettes... De nombreux exemples illustrent chaque chapitre.
Ref. : ML 153
Prix : **129** FF

LE LIVRE DU BASIC
Ce livre est une introduction complete et didactique au BASIC de l'ATARI ST. Tous les stades de la programmation sont couverts comme l'analyse du probleme, la realisation d'un ordinogramme, les differentes techniques de programmation... Le lecteur abordera ensuite des domaines plus pointus mais passionnants comme la programmation des graphismes, la gestion des tableaux a plusieurs dimensions, les techniques de tri, la gestion des donnees, le BASIC.
Ref : ML 154
Prix : **149** FF
NOUVEAU

DU BASIC AU C
Vous voulez vous mettre au C ? Pas de problemes! Ce livre s'adresse a tous ceux qui ont une bonne connaissance du BASIC et qui veulent se mettre rapidement au langage C. Cet ouvrage leur permettra de transferer leurs connaissances acquises pour le BASIC au C et ainsi de rentabiliser leur investissement ! En une journee ils pourront déja acquerir les fondements de la programmation dans ce super langage qu'est le C.
Ref. : ML 155
Prix : **149** FF
NOUVEAU

BIEN DEBUTER AVEC L'ATARI ST
Ce livre s'adresse a tous ceux qui démarrent avec le ST et qui veulent réussir à coup sûr leurs débuts ! Ils pourront grâce a cet ouvrage eviter les premiers pièges et profiter immédiatement de leur nouvelle machine. Domaines couverts : installation du matériel, utilisation du TOS et de GEM, la souris, l'écran, le clavier, premiers programmes en BASIC, le langage LOGO, les différentes possibilités d'utilisation du S, etc.
Ret. : ML 156
Prix : **129** FF
Disponible en Mai
NOUVEAU

8 9 10 11 12 — MICRO APPLICATION — ATARI ST

GRAPHISMES & SONS
Avec ce livre vous allez pouvoir profiter pleinement des immenses capacites graphiques et sonores de l'ATARI ST. Ce livre est accompagné de nombreux programmes exemples et utilitaires écrits en BASIC, LOGO, C et MODULA 2. Quelques-uns des thèmes abordes : graphismes en 2 et 3 D, graphismes sous GEM, les rotations, les figures moirees, trucs et astuces du graphisme.
Ret. : ML 157
Prix : **149** FF
NOUVEAU

LE LIVRE DU LOGO
Le LOGO est un langage très intéressant dont les applications sont tres nombreuses. Cet ouvrage permettra au lecteur de profiter au maximum du LOGO livre avec l'ATARI ST. Principaux thèmes abordes : les graphismes, les procedures, les récursions, les routines de tri, un generateur de masque, structure des donnees, intelligence artificielle.
Ref. : ML 158
Prix : **149 FF**
Disponible en Mai
NOUVEAU

LES LOGICIELS DE DESSIN SUR ATARI ST
L'ATARI ST a de fantastiques capacites graphiques et plusieurs logiciels du commerce permettent de les exploiter et de realiser de tres beaux dessins. Pourtant ces programmes ne sont pas toujours tres faciles a utiliser. Cet ouvrage vous dira tout sur GEM-DRAW, DEGAS et NEO-CHROME et vous livrera beaucoup de trucs et astuces qui vous faciliteront la tache. Vous pourrez ainsi tirer partie au maximum de votre logiciel.
Ref : ML 159
Prix : **129** FF
NOUVEAU
Disponible en Juin

LES GRAPHISMES EN 3 DIMENSIONS
Avec ce livre vous allez acquérir une connaissance complète de la programmation des graphismes en 3 D et de leur animation sur ATARI ST. Cet ouvrage vous enseignera tout d'abord les bases théoriques nécessaires à l'animation en 3D, puis vous expliquera clairement les notions de programmation en ASSEMBLEUR indispensables pour obtenir des résultats spectaculaires.
Prix : **179** Francs.
Réf. : ML 167
NOUVEAU
Disponible en Juin

DEMANDEZ LE CATALOGUE GRATUIT ! LIVRES ET LOGICIELS AMSTRAD, ATARI, APPLE, COMMODORE, 64 et 128, MS DOS, MSX, THOMSON...



DISQUETTES ATARI
Tome 3 — 120 F
Tome 6 — 120 F
Tome 9 — 120 F
Tome 12 — 120 F

POUR TOUTE COMMANDE, 1 MICRO-INFO GRATUIT !
Dans la limite des stocks disponibles.

MICRO APPLICATION
13, rue Sainte Cécile 75009 PARIS
tél. (1) 47-70-32-44

BON DE COMMANDE

DÉSIGNATION	QUANTITÉ	PRIX
	TOTAL TTC	

CB date d'expiration: ______

☐ Mandat ☐ Chèque ☐ CCP.
Libellez vos chèques à l'ordre de Micro-Application.
Nom, Prénom ______
Adresse ______
Ville ______ C.P. ______
CB ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐
Date et signature
+ 20 F de frais d'envoi
ou 40 F pour envoi recommandé.
Port gratuit pour toute commande supérieure à 250 F.

ST-05-86

EDITORIAL

LONDRES... HANOVRE... ...LA VILLETTE

Trois salons, de nouveaux matériels et une foule de produits qui symbolisent bien l'essor européen de la ligne ST. Nous avions pu déjà nous en rendre compte l'hiver dernier, mais la lecture de la presse française et de ses arrêts assassins à propos du ST nous faisait quelque fois douter de ce que nous avions vu.

Cette fois-ci, tout le monde est sur la même longueur d'onde, au moment ou le nouvel ATARI FRANCE fait un effort de publicité encore jamais vu pour la marque en FRANCE (Festival du Son, La Vilette, SICOB) (Le Monde, Le Figaro, Libération,...) et des pages en couleur dans tous les magazines spécialisés (SVM, OI, SOFT et MICRO, TILT,....).

Du coup, dans les rédactions on tourne sa veste avec aplomb et la machine dont l'achat était risqué il y a deux mois fait les couvertures aujourd'hui !

Ce n'était donc qu'une question de pognon, chapeau !

Ph. Giudicelli

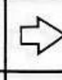
Comité de rédaction : Philippe GIUDICELLI. Jean-Michel DUBOIS. Godefroy GIUDICELLI. Directeur de la publication : Godefroy GIUDICELLI. Rédacteur en chef : Philippe GIUDICELLI. Directeur technique : Jean-Michel DUBOIS. Ont collaboré à ce numéro : J. NAVIA - A. NAJAND - M. ESSOUKAN - E. JESOVER - O. HARD - C. BONNET - P. FONTAINE - C. DROIN - C. BEUDIN - S. CARCONE - F. MARTON... et les autres.
ST MAGAZINE est édité par PRESSIMAGE. S.A.R.L. de presse au capital de 2.000 francs. 210, rue du Faubourg St-Martin, 75010 PARIS. Dépôt légal : 2e Trimestre 86. Tarif de l'abonnement : 200 francs (10 numéros).

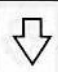
SOMMAIRE

AVRIL 1986

HANOVRE... LONDRES... LA VILETTE... LES SALONS DU ST 4
INTELLIGENCE ARTIFICIELLE 8
3 K DE FORTH MAJEUR 12
LA PUISSANCE DU 68000 12
QUELQUES CONSEILS POUR DEVELOPPER SUR L'ATARI 520 ST 14
COMMENT GERER GEM-VDI ET GEM AES EN BASIC 18
TROIS DIMENSIONS EN BASIC 21
OTHELLO/REVERSI EN BASIC 25
PROGRAMMATION SUR MFP 68901 28
PROGRAMMER EN ASSEMBLEUR : DECODAGE DU CLAVIER 33
MONTRE EN C 36
GEM : A QUOI CELA RESSEMBLE 38
NOUVEAUX PRODUITS 40
BANCS D'ESSAI 44
BANDE DESSINEE 46

Le présent numéro contient 8 pages couleur, non foliotées, intitulées « catalogue des produits de la gamme ATARI ST », encartées au centre. Elles sont offertes par MICRO-VIDEO. Ce numéro ne peut être vendu sans elles.





HANOVRE... LONDRES...



Après un automne chaud, les Salons ont recommencé à ouvrir leurs portes de plus belle en ce début de printemps. Début Mars, le Novotel de Londres accueillait exclusivement les Atariphiles pour le premier ATARI Show anglais de tous les temps. Quelques jours plus tard, la 'kolossale Hanover Messe' permettait à ATARI ALLEMAGNE d'étaler sur 650 m² la satisfaction d'une société qui a vendu à ce jour plus de 50. 000 ST. Enfin, pendant le Festival du Son, ATARI FRANCE avait squatté quelques pavillons du Grand Hall de La Vilette, et accessoirement quelques pages des grands quotidiens nationaux pour faire passer le message : « Atari est vivant, et aussi à Paris ».

Par ordre alphabétique des exposants, les meilleurs moments de ces trois manifestations. Vous retrouverez également certains de ces produits dans les pages de nouveautés.



Red Alert

--------- LONDRES : Tout pour la programmation -------------------

ANTIC est un groupe de presse californien qui s'est lancé dans la production de logiciels. Déjà connu sur ST pour **Mom and me, Muyrray and me** programmes conversationnels (en anglais !), **C. O. L. R. Object Editor** un utilitaire graphique passionnant et **Maps and Legends : the cartographer** un outil de pédagogie géographique. Ses nouvelles productions : **CAD-3D** un logiciel de dessin et d'animation tridimensionnel qui confirme son auteur, Tom Hudson, comme l'un des plus talentueux spécialistes du ST ; **Red Alert**, un jeu de simulation de guerre nucléaire dans la lignée de Wargame ; **Expert Opinion**, la version anglaise de L'Expert (cocorico !), le système expert de Mindsoft ; **Flash**, un programme complet de communication et **Star Struck** une incursion osée dans le monde de l'astrologie et des sciences divinatoires.

AST fait du hard pour ST et n'a rien à voir avec le fabriquant de cartes additionnelles pour IBM PC. Il offre pourtant des cartes 512K pour ST anémiques et des lecteurs 3'1/2 et 5'1/4 cpmpatibles avec la toute la gamme Atari 16 bits.

ATARI a fait un tabac avec ses nouvelles machines 1040 et 520 STF, mais surtout son émulateur IBM PC, montré ici en avant première de l'annonce officielle faite à Hanovre. (Voir News).

ATARI USER est un journal consacré aux petits ordinateurs de la gamme Atari. Il consacre de plus en plus de

PARIS



pages au ST, mais (d'après son directeur) n'ose pas encore faire une édition spéciale pour le ST. Heureux français qui ont eu le premier journal au monde consacré au 520 et à sa famille !

CASHLINK aura du mal a exporter ses produits. La comptabilité anglaise, çà peut vraiment servîr que là bas.

CHIPSOFT a le même problème avec son **ST Accounts**, un bien beau produit pourtant.

COMPUTER CONCEPTS revendique la paternité du premier Basic qui exploite vraiment la puissance du 68000. Nous n'avons pu savoir le fin mot, mais son air de famille avec le Basic BBC nous fait penser qu'il peut s'agir du Basic primitivement annonçé par une compagnie appelé BTMSOFT. **FAST ST BASIC** aura entre autre avantage le fait d'être disponible sous forme de cartouche. Pas avant Juillet cepandant. **EXECON LIMITED** a un peu plus de chances avec sa ligne complète de produits de gestion. Il y a surement quelques trucs à récupérer.

GST est le roi de l'éditeur. Son éditeur sous GEM est surement ce qu'il y a de mieux dans ses différents outils, C et Assembleur.
Il en a profité pour en faire un bon traitement de texte, **First Word,** qui circule de çi de là depuis la version 1. 0 jusqu'à la version 1. 6. On ne sait malheureusement toujours pas à quelle version ses éditeurs se décideront à le sortir. Une version 2. 09, entraperçue à La Vilette fait craindre le pire quand à la date effective de sortie. Cette dernière mélange à loisirs des textes et des graphiques issus de Neochrome et Degas. Suite du feuilleton au prochain numéro.

HABA SYSTEMS est une filiale de la boite américaine. Elle pourrait coiffer tout le monde au poteau quand à la mise sur le marché du premier traitement de texte sous GEM entierement françisé. Pour les autres **Habadex** et **Habaview qui** gèrent des fiches, la bataille s'annonce plus rude.

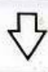
KUMA est connu pour son assembleur **K-Seka** et son tableur **K-Spread.** Ses nouveaux produits son essentiellement un logiciel de communication **K-COMM** et un programme de graphisme orienté gestion, qui peut d'ailleurs utiliser les données de **K-Spread** et du futur **K-Data.**

LASERSOFT Produit populaire sur le Macintosh, **Laserbase** est une base de données puissante, permettant de gérer un fichier jusqu'à 16 millions de caractères. (Voir News) Lasersoft produit également des accessoires de bureau, les **Power-pak. Diary, Adress book** et **Database** sont les premiers disponibles. **LLAMASOFT** est le créateur de l'innénarrable **Colourspace.** Si il a enthousiasmé un journaliste d'Hebdogiciel et également notre directeur de la publication, je connais pour ma part des démos du domaine public qui me semble plus riches. **MEGAMAX** est certainement l'auteur du produit le plus riche du Salon. Un outil de devellopement complet **Megamax C,** qui mérite le nom de Turbo C. Ses temps de compilation sont en effet 4 fois plus rapides que ceux du C du kit de dévellopement ATARI, pourtant déja performant et son code prend deux fois moins de mémoire. Il contient également un utilitaire puissant 'Resource Construction Program' qui permet de créer des menus déroulants, des boites de dialogue et des icones.

METACOMCO est une société prolifique dans les outils pour le 68000 (QL, AMIGA et ST). Son **Macro Assembler** est maintenant livré avec un linker (? ? ?), un **Lattice C** et un



Animate



Pascal sont maintenant disponibles, un **Lisp** viendra heureusement compléter la gamme.

MICRODEAL après un départ en fanfare (**Lands of Havoc,** premier jeu jamais publié pour ST) a vu son imagination brusquement se tarir. C'est sans doute pourquoi il distribue les produits de la société US Michtron, qui, elle, fait dans le genre famille nombreuse. Au milieu de produits moyens se glissent quelques bijoux. **Time Bandit**, par exemple établit un standard à battre dans le jeu d'arcade, titre auquel ne pourront prétendre ni **Major Motion,** ni **Gold Runner.** Dans le domaine des utilitaires **Dos Shell** pompeusement intitulé 'MS-DOS emulator' frise l'escroquerie, mais **Kissed** se révèle un petit bijou pour les amateurs de débuggage.

PAPERLOGIC nous était inconnu jusqu'à ce jour. Il n'avait pas de stand, mais il était difficile d'échapper aux confettis qu'il distribuait à la porte du Salon. **ST TOOLBOX** ne semble pourtant pas devoir bousculer les leaders dans l'utilitaire de disquettes.

PHILON , américain d'origine y va de son petit **PHILON (UK)**. Il en profite pour sortir ses premiers langages pour le ST. Spécialiste du compilateur pour grosse machine, ses rejetons Atari lui ressemblent comme deux gouttes de thé (England oblige) Ce sont des produits sérieux, bien débuggés, mais qui ne se sont pas encanaillés assez pour aller fréquenter GEM. Comme par les temps qui courrent, il y a intérêt à compiler rapide, tous ses produits commencent par Fast quelque chose. La famille Philon se décline donc comme suit : **FAST/Basic-M, FAST/C, FAST/Cobol, Henry's Basic** (Tiens ! Surement un batard.), **FAST/Fortran.** Blague à part, un compilateur Basic sur ST, çà peut servir. **PROSPERO** est un peu l'équivalent anglais de PHILON. Des langages évolués, avec comme premier produit un **Pro Fortran-77**, sorti le premier jour du Salon.

PSYGNOSIS avec son **Brataccas** dispute à **Sundog** le titre de meilleur jeu actuel sur ST. Non content de çà, ils en préparent un autre, encore plus géant, les bougres !

S. E. C. S est une émanation de Software Express, un distributeur anglais très spécialisé sur Atari. Ils ont inondé la presse avant le Show à propos de ce qu'on allait voir ce qu'on allait voir et pis tout çà. Ils ont surement des tas d'idées, mais çà doit être resté à l'état d'idées vu que ce n'est qu'au huitième tour du Salon, juste au moment de partir, que je me suis rendu compte qu'il présentait quelque chose sur le stand de ... Software Express. Ca a rapport avec la musique, (il y avait un synthé), mais quand j'ai demandé



une fiche technique, ils m'ont tendu une carte de visite. Ou mon anglais est tombé bien bas, ou ils sont pas vraiment prèts. Enfin, avec la carte, je sais toujours ou m'adresser pour en savoir plus ultérieurement.

------------ Hanovre : CAO et Analogique/Digital ----------------

ATARI ALLEMAGNE est actuellement le leader mondial de la vente du ST. Jack Tramiel ne s'est pas privé de dire tout le bien qu'il pensait de son représentant, le génial Dr STUMPFE. On pensait que, eu égards à l'importance du marché européen en général et du marché allemand en particulier, Atari profiterait de Hanovre pour montrer un certain nombre de nouveautés en matériel. Las. A part l'émulateur IBM PC, déjà vu à Londres, pas le moindre nouveau périphérique (imprimante Laser ?), pas de cartes (graphisme 1024x1024), pas de nouveau systèmes d'exploitation (Unix ?), pas de CD Rom, pas de nouveau chip musical. Seulement le 1040, 1 mega de Ram pour moins de 10. 000 francs en configuration complète, la routine déjà ! ! ! **CDS** propose différents logiciels et accessoires de création et de lecture de bar codes.

DATA BECKER met sur le marché plus d'un nouveau bouquin par mois concernant le ST. Ca nous intéresse au plus haut point, car MICRO APPLICATION son représentant en France les

Emulcom.

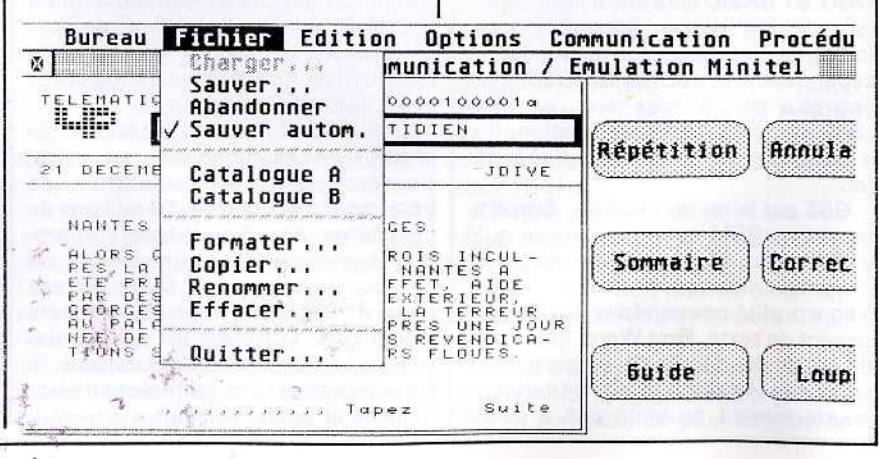
traduit au fur et à mesure. A venir (Sans garantie pour le titre exact) **ST pour débutants, BASIC sur ST, LOGO sur ST, DU BASIC au C sur ST, CP/M sur ST, ST Peeks et Pokes, ST Graphique et Son, Draw/Neo/Degas, Graphisme 3D sur ST, Le livre du floppy, Le ST pour tout le monde, L'intelligence artificielle sur ST.** Avis aux collectionneurs !
D'autre part **TEXTOMAT, DATAMAT** et consort ne devrait plus tarder non plus. Achh ! Doktor Becker !

MIR s'inscrira certainement dans l'histoire du ST comme le premier fournisseur d'un logiciel sur cartouche. **ST ime** permet de conserver l'heure, même après extinction du ST. Du pain béni pour les programmeurs.

OSS le plus grand spécialiste américain d'outils pour Atari, avec le légendaire Bill Wilkinson, propose un Pascal de très grande facture. 300 pages de doc pour utiliser un des langages de choix du ST.

PRINT TECNIC est le spécialiste allemand de la digitalisation vidéo. Deux produits, standard et pro, devrait bientôt faire leur apparition en France.

STEINBERG , connu pour ses produits musicaux professionnels, a fait un tabac avec **TWENTY FOUR** au Salon de la musique à Francfort. Destiné aux ST, il les transforme en séquenceur et magnetophones 24 pis-



tes. **TDI** est un spécialiste des cartes analogiques/digitales, mais son logiciel de recopie d'écran couleur du ST attirait particulièrement l'attention.

VERSASOFT Venu tout droit de Californie, pour montrer son produit vedette, le pdg de Versasoft a ajouté sa pierre à l'image de bon gestionnaire du ST. **Dbman** est en effet un clone de Dbase III. Il montre de plus d'intéressantes innovations par rapport à son modèle. Et surtout, son prix sera compris entre 1600 et 1800 francs.

La Villette : Comment transformer un abattoir en pressoir à pommes -

ADI est plus un prestataire de services qu'un distributeur de logiciels. Il propose un outil de developpement permettant d'écrire des programmes utilisant GEM et une structure de données **Adimens.**

ANDROMEDA a des gestionnaires californiens, des programmeurs hongrois et ils ont tout compris du graphisme du ST. A tel point qu'Atari leur a commandé ferme 8 jeux d'arcade dont trois sont prêts : **Battlezone, Cristal Castles** et **Millipède.** Mais ils étaient venus présenter leurs produits à eux : **ANIMATE** qui permet de faire ses propres animations et vous fera piétiner Movie Maker. **ST ART** qui calcule et reproduit tout objet dans toutes les dimensions et sous toutes les perspectives. **ATARI** a dans ses cartons plusieurs produits, dont certains depuis longtemps (**Chiffres et lettres**) Esperons la disponibilité prochaine de **1st Word** et surtout d' **Emulcom.** Rendez votre Minitel Intelligent pourrait être le sous titre d'Emulcom. (Voir News)

GUDULE est une société sérieuse, si quelqu'un en doutait. Ses deux premières productions **PI DISK** est un accéssoire de bureau qui charge automatiquement une unité de disque virtuel avec son contenu. Le temps d'accès disque est multiplié par 200. **PI COMPTA** est une comptabilité qui contient un tableur et permet 350 comptes, 15 journaux généraux et 2500 écritures par période. (10000 par an)

HIPPOPOTAMUS Beaucoup de nouveaux produits chez cet éditeur américain, parti très tot dans la course sur ST. **Hippo word** promet d'être un des tous meilleurs traitements de texte sur le ST. Il supporte des fonctions évoluées comme l'édition multi-colonnes, des statistiques sur les mots, le mixage graphique/texte et des polices de caractères nombreuses. Il est rendu encore plus puissant par des logiciels accéssoires : **Hippo Fonts** qui lui donne de nombreuses polices supplémentaires, **Hippo Pixel** qui permet de créer des polices si celles pré-existantes ne suffisent pas et **Hippo Laser** qui permet de gérer une imprimante laser.

INRETS est l'auteur d'**Astek**, « un programme de communication qui permet à l'utilisateur du ST de l'employer comme un terminal graphique monochrome Tektronix connecté par ligne asynchrone sur tout ordinateur » Là, j'ai recopié mot à mot la brochure parce que j'ai pas tout compris. Généralement dans ces cas-là, quand on cherche à résumer, çà veut plus rien dire du tout.

LORICIELS est désormais le représentant français d'Activision. A ce titre on pouvait voir **Borrowed Time, Hacker, Mindshadow** et le premier logiciel musical sur ST, **Music Studio,** qui gère le chip interne et des synthétiseurs externes.

MECANORMA NEMOGRAFIC Bien sur avec un nom pareil, c'était eux les plus chers du Salon. Il faut rentabiliser le papier à lettre. Pour 29. 900 francs (Hors Taxe !) ils proposent un logiciel 2D/3D pour architecture, génie mécanique, électronique ... . A noter que pour ce prix là ils fournissent également la machine. On pourrait croire le logiciel particulièrement complet, mais si l'on en croit leur documentation, le transfert de données (?), le programme de calcul (? ?) et la quantification (? ? ?) sont en option. On rigole, mais c'est surement très bien.

MEMSOFT C'est signé ! Avec conférence de presse à l'appui pour montrer l'interêt porté par chacun à l'accord MEMSOFT/ATARI. Aux termes de cet accord, le 1040 disposera bientot de MEMDOS et par conséquent des dizaines de logiciels professionnels qui tournent avec. Oui ! l'Atari a un grand avenir et d'énormes perspectives sur le marché professionnel. Merci Mr Tramiel ! Oui, Memsoft a une des bibliothèques les plus riches en progiciels de toute sorte, horizontaux et verticaux. Merci Mr Lafitte !
Les deux fiancés se poursuivaient depuis des mois. C'est chouette les histoires d'amour qui finissent bien.

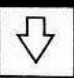
PRESSIMAGE Eh oui ! On est aussi de la fête. En plus des logiciels de la société Gudule, nous sortons **ST Fact,** un programme de gestion de stocks, de gestion clients et de facturation.

PROGISERVICES propose **Progicompta,** une comptabilité générale avec possibilité d'associer une comptabilité analytique, une comptabilité budgétaire et une édition du bilan.

A signaler deux applications musicales, ne portant pour l'instant les couleurs d'aucune société, un logiciel de création musicale tout à fait original sur lequel nous reviendrons et le ST comme accéssoire (sic !) d'un Yamaha DX7.

Signé RDS

Prochain rendez-vous du Rodeur des Salons (RDS) : C. E. S. de Chicago



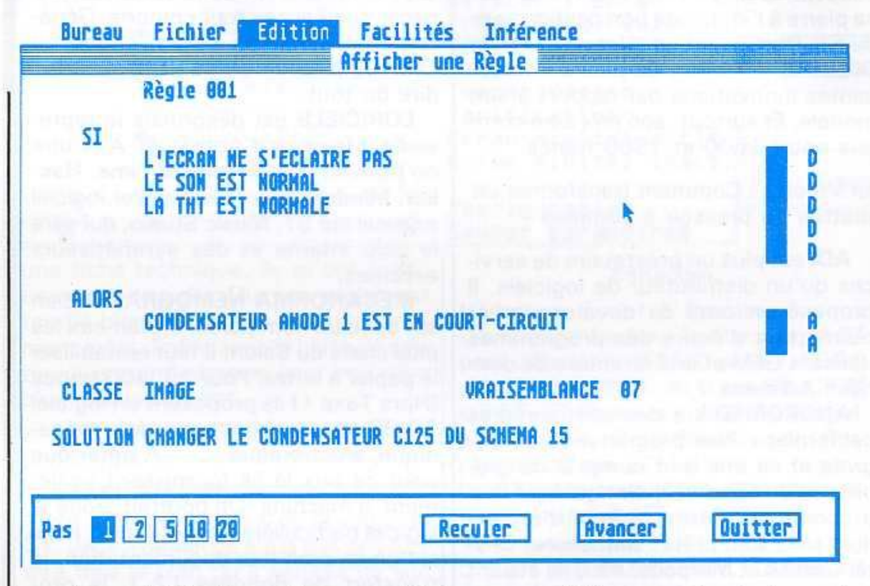
INTELLIGENCE ARTIFICIELLE

SYSTEMES EXPERTS : Outils d'aide à la décision ou raccourci pour l'Intelligence Artificielle ?

On peut définir un « Expert » comme une personne qui a plus de connaissances sur un sujet donné que la moyenne des individus. Si vous ou quelqu'un de votre famille est malade, vous irez probablement voir un médecin. Après avoir posé plusieurs questions ou effectué différents tests, le médecin parvient à un diagnostic et prescrit un traitement. Si le moteur de votre voiture s'arrete intempestivement, vous irez certainement voir un garagiste qui fera des vérifications et les réparations nécessaires pour la remettre en état. Si vous vous apercevez que vous payez trop d'impots sur le revenu, vous demanderez probablement conseil à votre expert comptable ou conseiller fiscal.
Tous ces « Experts » que vous consultez, sont considérés comme des personnes ayant une bonne connaissance de leur domaine d'expertise, leur compétence peut ainsi nous permettre de résoudre rapidement notre problème.
Aujourd'hui, il existe des programmes appelés Systèmes Experts capables de se comporter comme ces « Experts » dans les domaines d'expertise pour lesquels ils ont été construits (médecine, électronique, finance, etc...).
Ces programmes font partie des premiers résultats importants en Intelligence Artificielle.

Est-ce la fin des « Experts » ?

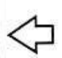
Le but recherché par ces programmes n'est pas de remplacer les « Experts », mais de les multiplier, d'augmenter leur productivité en les rendant plus disponibles pour les taches mettant en œuvre leur créativité, de mettre leurs connaissances à la disposition d'un grand nombre d'utilisateurs.
Les « Systèmes Experts » se composent d'une Base de Connaissances du domaine d'expertise qui regroupe les connaissances d'au moins un « Expert humain », d'un moteur d'inférence qui représente le mécanisme d'utilisation de ces connaissances et d'une Base des faits qui va contenir la spécification du problème à traiter et les résultats trouvés par le système expert. Ces systèmes entièrement fondés sur la connaissance se caractérisent par leur souplesse d'utilisation, et leur possibilité d'expliquer COMMENT ils sont parvenus à un résultat, ou POURQUOI ils ont besoin de certains renseignements à un moment donné, mais aussi par leur capacité de donner une solution adaptée à chaque cas.
Par ailleurs, le fait que la base de connaissances et les échanges avec le système soient en langage naturel assure une modification et une mise à jour aisées des connaissances du système. L'ordre d'entrée des connaissances n'influe pas du tout sur les résultats et dans certains cas le système peut meme trouver des résultats en l'absence d'une description précise du problème.



CONNAISSANCES + INFERENCES = SYSTEMES EXPERTS

Mind Soft 3, rue de l'Arrivée BP 63
75749 Paris cédex 15
Tél (1) 45. 38. 70. 12

L'intelligence artificielle démarre vraiment comme application personnelle de la micro avec le ST. Avant lui, les machines abordables étaient trop peu puissantes, les machines performantes hors de portée de la bourse d'un particulier.
Dans ce domaine tout est à faire. Ce mois-ci, nous vous avons présenté succintement les systèmes experts à l'occasion de la sortie de l'Expert de Mindsoft pour l'Atari ST. Pour 1.490 francs, vous pouvez rentrer dans le monde de l'intelligence artificielle, en fabriquant votre propre base de connaissances dans la matière dans lequel vous êtes... « expert ».
L'Expert a été développé sur Macintosh, IBM PC et compatibles et sur Apple II. Les bases de connaissances peuvent être échangées d'une machine sur l'autre, ce qui assure à votre application des débouchés sur un grand nombre de machines.
Pressimage et ST Magazine sont intéressés à l'édition de toute base de connaissances présentant un intérêt pour les autres utilisateurs. Elles pourront être commercialisées soient à destination de ceux qui possèdent le système expert de Mindsoft, quel que soit la machine, ou encore sous forme de « run time », à n'importe qui intéressé par ce domaine d'expertise.
Pressimage récompensera par des abonnements et des cadeaux divers toutes bases de connaissances menées à leur terme, et éditera en collaboration avec Mindsoft les meilleures d'entre elles.
L'intelligence artificielle représente une des voies les plus excitantes de la micro dans les prochaines années. Que ce soit des applications d'intelligence pure ou l'adjonction d'une certaine intelligence à des applications classiques, le champ est vaste pour exercer vos talents. ST Magazine couvrira dans chaque numéro toute l'actualité dans ce domaine, en espérant que pour une part, l'actualité ce sera vous. A vos machines !





ENTREZ DANS LE MONDE DE L'ATARI ST
avec RUN DISTRIBUTION
62, rue Gérard - 75013 PARIS
Tél. (1) 45.81.51.44 - Télex RUNINFO 270841 F
• DE VRAIS UTILITAIRES
• TESTÉS ET SÉLECTIONNÉS
• DES OUVRAGES (français et étrangers)
• UN SOUTIEN AVANT ET APRÈS-VENTE
• DES PÉRIPHÉRIQUES
• DES PRIX
• LIVRAISON SOUS 48 HEURES
2 exemples :
USA
16/32 bits
menus déroulants
à peine sortis déjà disponibles !!!
souris
Haba
Des programmes qui utilisent pleinement les capacités du ST
icônes
sous GEM
1 LA GAMME
HabaView
FOR ATARI ST
Version Française
French Version
PERSONAL INFORMATION MANAGER AND PHONE DIALER
HABAWRITER MAIL MERGE UTILITY
...à suivre !!
HABAWIEW
• Une base de données puissante, souple, intuitive.
• Intuitive car HABAWIEW fonctionne sans apprentissage spécial.
• 32 champs par fiche (6 types de champs : ALPHANUMÉRIQUE, DATE, OUI/NON, NUMÉRIQUE, HEURE, ALPHABÉTIQUE).
• Modification instantanée des informations.
• Possibilités d'occulter des colonnes à l'écran et à l'impression.
• La configuration en colonnes de la base de données génère automatiquement le masque de saisie.
• Impression des fichiers avec HABAWRITER (coller) ou en fichier ASCII.
• Impression d'étiquettes.
• Tri simple ou progressif.
Prix : 895 F
HABAWRITER
• ENFIN UN TRAITEMENT DE TEXTE EN FRANÇAIS.
• TRÈS FACILE D'UTILISATION (orienté utilisateur).
• Affichage à l'écran et impression sur imprimante, des caractères accentués, des caractères GRAS, soulignés, indices, exposants.
• Fonction couper-coller, frappe au kilomètre, césure.
• Reconfiguration du clavier (allemand, espagnol, etc...).
• 6 fenêtres disponibles à l'écran permettent de travailler sur un texte tout en consultant 5 autres documents.
• SAUVEGARDE DES DOCUMENTS en mode HABA ou ASCII.
Prix : 895 F
HABAMERGE
• Un puissant logiciel de fusion de fichiers.
• Liaison simple entre une liste de fiches et un document.
• Affichage d'un état pendant l'impression des fiches fusionnées.
Prix : 490 F
HABADEX
• Gestionnaire de fichier simple et performant.
• 300 fiches de 16 champs par dossier.
• Tri rapide.
• Edition d'étiquettes.
• Retrouve instantanément une fiche donnée.
• Se relie à HABAWRITER grâce à HABAMERGE.
Prix : 740 F
HABA...
utilitaire de communication
menus déroulants
16/32 bits
icônes
sous GEM
souris
2 LE DISQUE DUR HABADISC POUR ST
• Stockage et recherche des données à une vitesse éclair.
• Disque DUR WINCHESTER, stocke 10 MB (équivalent à 27 disquettes ATARI SF).
• Vitesse de transmission de l'information : 5 millions d'octets par seconde.
• 345 pistes par pouce, 306 cylindres, 1124 pistes.
• Livré avec câble interface ATARI.
• Fonctionne avec moniteur couleur.
RAPIDITÉ ET PUISSANCE
Prix : 8500 F
DISTRIBUTEURS CONTACTEZ-NOUS
RUN DISTRIBUTION
62, rue Gérard 75013 PARIS
Tél. (1) 45.81.51.44 - Télex : RUNINFO 270841 F
Importateur pour la France RUN INFORMATIQUE
Bon à renvoyer à RUN DISTRIBUTION, 62, rue Gérard, 75013 PARIS
• FAITES-MOI PARVENIR DE TOUTE URGENCE LE DOSSIER HABA □
• TÉLÉPHONEZ-MOI □
• Matériel vendu : ....
• Je suis Grossiste □
• Distributeur □
Monsieur ....
Prénom ....
Fonction ....
Société ....
Adresse ....
Tél. .... Télex : ....

3 K DE FORTH MAJEUR

Que faire d'un nouveau langage, au milieu de tant d'autres comme le C, le Pascal ou le Basic ? Le Forth est pourtant bien à part dans cette jungle inextricable des langages.

Il est généralement admis que plus un langage est proche du code machine, plus il est rapide et moins il tient de place en mémoire. Et au contraire, plus il est abordable, comme le Basic, plus il est lent et vorace en mémoire. Le Forth est une exception à la règle : véloce et concis (il peut être aussi rapide que le C et même plus dans certaines conditions). Car le Forth tout comme le C, est un langage compilé, et structuré.

Les seuls concepts à comprendre pour programmer en Forth sont la pile et le dictionnaire :

- La **pile**, les programmeurs en assembleur la connaissent bien, pour les autres, on peut la comparer à une pile d'assiettes où au lieu d'avoir des assiettes, on aurait des chiffres que l'on manipulerait (rotation, duplication, suppression, etc...). Cette structure de la pile entraine que toutes les opérations mathématiques et logiques se font en polonais inverse (le « PRINT 2 + 2 » du Basic s'écrit « 22 +. », le point sert à lire le sommet de la pile, A = A + 2 du Basic donne, 2 A + !).
- Le **dictionnaire**, quand à lui, est composé, au chargement du Forth, d'un certain nombre de primitives. A partir de ces primitives on va pouvoir écrire de nouveaux mots qui vont se rajouter au dictionnaire et qui, eux aussi pourront être appelés dans de nouvelles définitions, et ainsi de suite. Cette notion de dictionnaire qui évolue suivant les besoins est ce qui fait la puissance du Forth et qui le rend adapté à n'importe quel travail, aussi bien la gestion, le calcul, que les jeux. De plus cette façon de programmer élimine pratiquement toutes les erreurs. Le programme est constitué de mots sans erreurs car on aura pu les tester séparément.

Mais avec cette puissance, qu'existe-t-il sur ST pour programmer en Forth ? A l'heure actuelle, il y en a trois : **le 4XFORTH de Dragon Group, Holmes** & Duckworth Forth de **Mirage Concept,** et le **UNIX FORTH** du **San Leandro computer club.**

Le 4XFORTH Level 1

Celui réservé aux développeurs comporte un source entièrement commenté, ce qui est très pratique pour modifier certaines des fonctions du dictionnaire de base.

Pour écrire des programmes il faut un éditeur, celui du 4X est très puissant, il permet d'effacer une ligne entièrement ou en partie seulement, de la mettre dans un buffet et de la replacer, là où on en a besoin. Le curseur peut être déplacé avec les flèches ou avec la souris. Il est même suffisamment puissant pour servir de mini éditeur de textes.

Il possède un RAM-DISK incorporé très pratique ; Le Forth travaille sur des petits fichiers tenant sur un écran ; cela évite les perpétuels accès au disque.

Pour les entrées-sorties il possède des instructions pour configurer la RS 232 C, mais c'est à peut près tout (c'est un peu léger pour un langage comme le Forth). Pas de gestion du port MIDI ou de la souris.

Sur le plan graphique il est assez bien pourvu, on y a adapté la LIGNE A du GEM (il trace triangles, diamants), et un certain nombre des fonctions du Tektronix 4010 (spécialisé dans le graphisme) comme MOV, DRW (déplacement et dessin) ou RMOV, RDRW (idem, mais relatif à un point).

Il possède un programme accessoire pour optimiser le code objet obtenu par le Forth : ACCELERATE. Il permet d'augmenter encore la vitesse d'exécution mais au détriment de la taille.

Mais l'énorme problème, le gigantesque problème de ce Forth est qu'il est complètement fermé et n'a aucun accès au GEM, ce qui est quand même une tare sur une machine comme le ST.

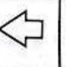
Holmes and Duckworth Forth de Mirage concept

Il est quant à lui de meilleur facture. La documentation est incorporée sur la disquette, il possède aussi une source commenté.

Son éditeur est pleine page mais succint, avec juste le nécessaire pour éditer des programmes. On peut remédier facilement à ce problème en y rajoutant certaines fonctions grâce au source.

Ce Forth permet aussi de gérer toutes les entrées, sorties comme le port MIDI (MIDIN, MIDIOUT), la souris ou le port parallèle centronic, et ceci avec des mots Forth.

Pour le graphisme, il a entièrement accès au GEM avec des instructions comme AESCALL, VDICALL ou bien encore GDOS.



Tant que l'on est dans le GEM, ce Forth permet aussi de gérer des fenêtres ou des menus déroulants grâce à des mots Forth.
Pour la gestion des chaines de caractères pas de problème, il a tout ce qu'il faut (TYPE, EXPECT...) et même des mots pour transformer les chaines de caractères du format C (avec un zéro au bout) en chaines au format Forth (avec au début le nombre de caractères).
La gestion des fichiers, elle aussi, est bien faite avec un mot permettant de se positionner à l'octet près dans un fichier ou de charger « n »'octets d'un fichier.
En plus de tout cela, il a également aussi des mots pour calculer sur des nombres réels à virgule flottante, ce qui est assez rare en standard sur un Forth. Il a même les instructions CASE pour gérer tout une suite de cas, dans un menu par exemple. (Cette instruction permet de remplacer avantageusement toute une série de IF THEN en BASIC).
Et pour finir ce Forth est entièrement relogeable (ORG) donc pas de problèmes pour les sous-programmes en assembleur.
Ah, j'oubliais ! Il permet de créer des programmes « click-on » qu'il ne reste plus qu'à « clicker » pour les voir s'exécuter.
Le seul gros défaut de ce Forth est qu'il est sensible aux majuscules, minuscules ce qui oblige à retaper un mot exactement de la même façon, qu'il a été défini.

UNIX FORTH

C'est peut être le moins standard de tous ces Forth mais c'est aussi le plus puissant.
Il a bien sur un source commenté. Mais sa grosse différence est que même le dictionnaire de base est modulable : la disquette est remplie d'utilitaires en tout genre que l'on peut charger ou non suivant les besoins.
Parmi tous ces utilitaires il y a un assembleur, un désassembleur, un programme de DUMP, et, oh miracle il y a même un décompilateur (vraiment très pratique pour récupérer le source de son programme).
A côté de cela il a entièrement accès au GEM (LINE A graphique BIOS...) avec SYSCALL.
Comme si ce n'était pas suffisant il permet de linker des programmes Forth avec des programmes C ou Assembleur. Il peut même exécuter des programmes autres que Forth puis revenir au Forth
En prime on peut gérer les cas (CASE), ainsi qu'une nouvelle pile en plus de la pile return et de la pile paramètre.
Ses seuls gros défauts, (car il en a) sont qu'il est sensible aux majuscules minuscules, et surtout qu'il ne possède pas d'éditeur pleine page, mais seulement un ridicule éditeur ligne à ligne.
En conclusion, le 4XFORTH (Level 1) de par sa configuration fermée est le moins bon des trois, mais, il semblerait que le Level 2 doive remédier à presque tous les défauts de son petit frère, attendons pour voir.
Le H&D est un bon outil de développement, et en connaissant le Forth, étant le plus standard des trois, on se sent en terrain connu.
Le UNIX est le meilleur des trois, mais cette qualité se paye : il est aussi le plus compliqué à mettre en œuvre. Il faut reconnaitre qu'il ne lui manque presque rien.
Au vu de la puissance des Forths existant sur ST, le Forth pourrait devenir un langage de développement assez répandu. De toute façon, les programmeurs intéressés ont déjà de quoi bien s'amuser.

Eric Jésover

LA PUISSANCE DU 6800O

Il existe actuellement deux courants : Le premier, Américain, assure que l'avenir est au 16 bits. Le deuxième, Anglais, tend quant à lui à penser que le 16 bits n'apporte rien à la micro familiale et que le 8 bits a de beaux jours devant lui. Jetons donc un regard à ce nouveau processeur 16 bits qui est en train de s'imposer sur le marché Américain...

La puissance du ST réside principalement dans le microprocesseur MC68000 de Motorola, processeur qui est très vite devenu le standard de la nouvelle génération de Super Micros (Macintosh, Amiga, 520ST ...). Car plus qu'une puce à la mode, comme le fut il y a quelques années le célebre 6502, le 68000 se révèle être un choix logique pour quiconque cherche à produire une machine véritablement puissante.
Un système proprement conçu autour d'un 68000 (et c'est le cas du ST) possède suffisamment de puissance pour supporter à son avantage la comparaison avec des ordinateurs référencés à la fin des années 70 comme « mini ordinateur » ! La comparaison est d'autant plus significative que sur bien des points le 68000 a été conçu spécialement pour reproduire de tels systèmes sur une puce...
Un Mini type, consiste en une unité centrale et un petit nombre de terminaux, chacun capable d'exécuter séparément des programmes différents et ceci au même moment. Chaque utilisateur d'un terminal possède un mot de passe qui lui donne un accès limité au système. En pratique, le processeur central alloue un certains temps d'exécution à chaque programme, temps qui dépend du nombre d'utilisateurs et de leur degré de priorité. Il existe en effet une hiérarchie des utilisateurs et des programmes : En règle générale, c'est souvent le programme qui fait le plus d'entrées/sorties qui est le plus prioritaire.

Les systèmes multi-postes.

Un terminal peut être défini comme celui du Superviseur. On peut alors à partir de celui-ci prendre complètement le contrôle du système et définir qui peut faire quoi, allouer plus de mémoire à tel utilisateur et moins à tel autre, empêcher l'accès à certains périphériques, etc...
Et le 68000 dans tout cela ? Et bien le Motorola est lui même conçu pour rendre possible la fabrication de systèmes multi-utilisateurs à micro processeurs, autrement dit de micro-ordinateurs ***multipostes***. Mais comme tout processeur, a)très rapide, b)pouvant adresser suffisamment de mémoire, c)étant apte à reconnaitre beaucoup d'états spéciaux nécessitant un traitement immédiat (les ***Interruptions***), le 68000 peut être utilisé pour exécuter plusieurs programmes différents en même temps c'est ce que l'on appelle le traitement « ***multi-tâche*** ».
Le 68000 possède en plus deux modes différents :
- Le Mode Utilisateur (ou mode esclave),
- Le Mode Superviseur (ou mode maitre).
La plupart des programmes sont utilisés en mode esclave. Dans ce cas, le programme a seulement un accés limité au processeur. L'autre type donne un accés plus complet au 68000. Un système peut donc être conçu autour d'un unique 68000 avec plusieurs usagers et un superviseur possédant un contrôle complet.
Même pour l'unique utilisateur du ST, certaines zones de la mémoire peuvent uniquement être accédées en mode superviseur, protégeant ainsi le système, de toute violation par le logiciel en cours d'éxécution. De plus, les programmes en langage symbolique 68000, sont toujours écrits dans ce que l'on appelle « THE POSITION INDEPENDANT FORM », c'est à dire que le programme peut être chargé n'importe où dans la mémoire disponible et peut être éxécuté même si un autre programme occupe un autre espace de la mémoire.
Quelle est la relation avec l'utilisateur individuel du ST ? Et bien, en fait, elle est beaucoup plus importante qu'on pourrait le croire, puisque le ST est toujours multi tâche, même si cela n'est pas visible.
Les accessoires du bureau GEM, tels que la calculatrice, la montre, ou le breakout, sont en fait en exécution permanente. Dans la plupart des cas, il s'agit en fait pour le système de se renseigner si l'utilisateur est en train d'essayer d'accéder à l'un d'eux. Ce qui explique pourquoi ils peuvent s'exécuter durant le déroulement d'une quelconque application GEM.
Parallèlement à ceci, la complexité du système GEM est telle, que celui-ci est en permanence en train d'effectuer une multitude de tâches, sans pour autant nuire à la rapidité...
On le voit donc le 68000 est suffisamment puissant pour supporter un système multi-poste, multi-taches. La rumeur courante qui veut que le ST devienne d'ici peu une machine UNIX (Système d'exploitation mp-mt le plus connu) est donc loin d'être absurde. Cependant vu la taille de ce système seule une machine équipée d'au moins 1 Mega de mémoire pourra le supporter.

Plus de puissance.

La vitesse du ST est basé sur différent facteurs dont le nombre élevé de coprocesseurs. Mais, le plus important est que, contrairement aux microprocesseurs 8 bits comme le Z80 (TRS80, Sinclair Spectrum) ou le 6502 (Apple 2, Atari 800XL), le 68000 travaille par tranche de 16 bits.



Une des principales limitations de vitesse d'un processeur est le taux auquel les données y entrent et en sortent. Par conséquent en doublant le nombre de bits qui peuvent être manipulés simultanément, on accroit la vitesse d'éxécution des instructions. Ce n'est pas la seule façon de descendre les temps de commmunication. Comparé à la plupart des CPU, le 68000 économise du temps en séparant les systèmes en communication avec le monde extérieur, de ceux qui manipulent le materiel interne à la puce. S'il existe plusieurs instructions en attente à l'extérieur du processeur pendant qu'une opération s'éffectue à l'interieur de celui-ci, le système d'E/S permet l'empilement de 2 instructions prêtes pour une utilisation immédiate dés que la partie principale du processeur sera prête.
Une fois à l'interieur, les données sont manipulées à une allure très rapide : par exemple, un 8 bits comme le Z80 est capable d'éffectuer 4 millions d'opérations par seconde , quand le 68000 peut lui tourner au taux de 8 millions de cycles par seconde.
La différence ne s'arrête pas là car le 68000 n'est pas un simple 16 bits. Il est couramment répertorié comme un 16/32 bits ou « faux 32 bits ». Alors qu'il communique avec l'extérieur ***par mot*** de 16 bits, ceux sont des mots de 32 bits qui sont manipulés à l'interieur même de la puce. Ceci représente un avantage important. Prenons l'exemple d'une instruction du 6502 qui utilise une opération trop importante pour être entrée sur 8 bits et qui nécessite 32 bits. Le 6502 devra découper cette opération en 4 étapes distinctes alors que le 68000 peut lui, la manipuler en entier, d'où un gain de temps non négligeable.
Le 68000 posséde pas moins de 17 registres internes, même si 16 seulement peuvent être utilisés au même instant : 8 pour les données, 9 pour les adresses (8 seulement étant utilisés). Ceci ne représente pas uniquement un accroissement de puissance, mais un gain de temps, étant donné que plusieurs opérations, nécessitant differentes valeurs, peuvent être instantanément manipulées.
Une autre possibilité exceptionnelle du 68000 est de pouvoir reconnaitre et donc traiter 256 ***interruptions*** (en 68000 on parle généralement d'« exceptions ») différentes.

Une tâche urgente.

Chaque microprocesseur doit être apte à reconnaitre un certain nombre de fonctions urgentes, à suspendre instantanément l'exécution du programme courrant, et à trouver les instructions nécessaires au traitement de la tâche urgente. Un processeur comme le répendu 6502 ne reconnait que trois interruptions differentes (IRQ, RESET, NMI). Quand une interruption arrive le 6502 doit surveiller l'ensemble du système, interrogeant chaque interface, chaque controleur, pour savoir lequel a fait une demande d'interruption du programme. Le 68000 quant à lui, est capable d'identifier immédiatement d'où vient la requète.

La mémoire.

Enfin, la véritable supèriorité du 68000 (face aux autres processeurs de micros), réside dans l'espace mémoire adressable.
Nous avons décrit le 68000 comme un microprocesseur 16/32 bits, mais cette description doit être completée. En effet ***le bus d'adresses*** a une largeur de 24 bits, permettant au 68000 d'adresser directement 2^{24} octets de mémoire contre 216 pour la plupart des 8 bits : soit en fait 16. 777. 216 octets (=16 Mega) contre 65. 536 (=64 Ko), No Comment !
On le voit donc les 512 Ko de mémoire, montés d'origine sur le ST, c'est beaucoup comparé aux autres micros, mais bien peu comparé aux possibilités offertes par le processeur. Sachez cependant, que vous pouvez étendre assez facilement la mémoire de votre ATARI. Il existe déjà un kit d'extension à 1 Méga, mais le ST a été conçu de façon à supporter sans modification majeure 4 Mega de mémoire vive, soit 16 puces de 256Ko ! (c'est fantastique quand on pense qu'un gros ordinateur comme l'IBM 4341 posséde en version de base 2Mo de mev !).

Conséquence directe de cette fabuleuse puissance, le 68000 est relativement complexe à programmer comparé au simple 6502 ou même au Z80. Peut être dois-je dés maintenant avertir les débutants que, dans la mesure où les instructions de bases du 68000 peuvent très simplement s'associer, Il y en a en fait 57 341 ! Mais la rapidité de ce processeur est telle que l'on préfère généralement programmer dans des langages moins rapides mais plus évolués et plus simples d'apprentissage tel le Langage C.

Bien sûr, d'un point de vue strictement familial, de telles performances n'apportent rien, (pour le moment tout au moins), si ce n'est la satisfaction de savoir que l'on a sous les doigts une puissance que bien des gros ordinateurs auraient aimée posseder il y a seulement 10 ans. Mais avec la récente baisse de prix du 520ST, qui l'amène quasiment au même prix que le commodore 128 (ordinateur 8 bits), il n'y a plus à hésiter. Les Américains ont sur ce point raison, il est temps de passer à l'ère du 16 bits et Atari en est le grand témoin ! ...
****
****
****

DICTIONNAIRE :

Multi-tâche : On regroupe sous ce terme les concepts de multiprogrammation et de multitraitement : Cependant ceux sont deux concepts différent ! Contrairement à ce que certains écrivent, le 68000 ne permet pas le multitraitement mais uniquement la multiprogrammation.
En multiprogrammation à un instant t quelconque, Il n'y a qu'**un seul** programme en éxécution dans l'unité centrale.
En multi-traitement à ce même instant t il y a plusieurs programmes car on utilise plusieurs UC...

Interruptions : Les interruptions sont des demandes d'action, demandes prioritaires qui obligent le système à suspendre l'éxécution normal d'un programme, afin de la traiter par l'exécution d'une tâche prioritaire spécifique au type de l'interruption.

Mots et Octets : On parle d'octet quand une opération est codée sur **8 bits**, et de Mots quand elle est codée sur plus (16, 32, 64...).
Dans le langage 68000 on manipule des Octets, des Mots(16 bits), des Longs Mots(32 bits).

Bus d'adresses : Liaison permettant l'interconnection entre les mémoires et le processeur afin que celui-ci puisse spécifier l'adresse mémoire à laquelle il veut accéder. Les données contenues dans cette adresse sont renvoyées au processeur par le bus de Données.

Quelques mots sur UNIX : UNIX à l'encontre de l'actuel système du ST (CP/M68) est un sytème d'exploitation multi-utilisateurs, capable de traiter plusieurs tâches en même temps. UNIX est devenu aux Etats Unis un standard de fait pour les gros micros et les minis : gamme VAX, Micral 9050. Sa souplesse d'utilisation et ses capacités d'extension ont conquis les milieux industriels et scientifiques. Une des particularités de ce système est qu'il a été ecrit en Langage C...

QUELQUES CONSEILS POUR D

A l'heure actuelle un certain nombre d'outils de développement en langage C et assembleur sont disponibles sur l'ATARI 520 ST, outils dont la mise en œuvre n'est pas obligatoirement évidente pour un programmeur néophyte. Aussi voici quelques conseils afin d'éclairer quelque peu le chemin semé d'embuches du programmeur en question.

1 - CONFIGURATIONS MATERIELLES

Autrement dit ce dont il faut disposer au minimum pour pouvoir développer.

Configuration A (→ CA = minima) :
Un ATARI avec 520 Koctets de mémoire RAM (260 ST ou 520 ST)
Un lecteur de disquette 360 K (c'est nettement plus facile avec un lecteur de 720 K)
Un moniteur, de préférence haute définition et noir et blanc.

Configuration B (→ CB = confortable) :
CB = CA + un lecteur de disquettes double face (720 K).

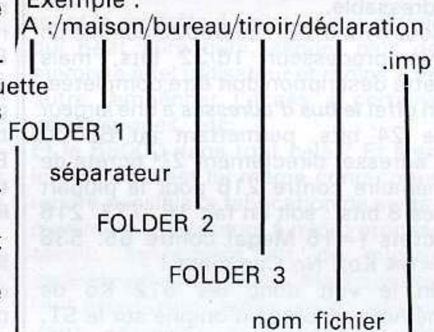
Configuration C (→ CC = de luxe) :
— Un atari avec 1024 Koctets de mémoire RAM.
— Deux lecteurs de disquettes dont un double face.
— Un moniteur noir et blanc haute résolution.
— Une imprimante (cher, mais très pratique).

2 - QUELQUES RAPPELS SUR LA MACHINE

Démarrage de la machine :
Après le chargement du TOS (disquette système que l'on a intérêt à sauvegarder par ailleurs), il reste environ 250 Kram disponibles dans une configuration type CA ou CB, ou 750 Kram dans une configuration type CC.
Ceci est important, car si un de vos programmes s'avérait trop gourmand en mémoire (vis-à-vis de la place disponible), vous risquez fortement d'avoir à réinitialiser votre machine après avoir admiré quelques effets graphiques inattendus sur votre moniteur.

Système d'exploitation :
GEM et CPM/68k.
Un système d'exploitation est constitué d'un ensemble de programmes permettant l'utilisateur d'accéder aux différentes fonctionnalités de la machine.
Il est utile (mais non indispensable) de se procurer la documentation concernant GEM et CPM, si l'on veut pouvoir bénéficier des caractéristiques graphiques de l'ATARI.

Organisation des fichiers :
Les fichiers sous GEMDOS peuvent être organisés hiérarchiquement en fonction de catégories définies par l'utilisateur des fichiers, plus prosaïquement, l'utilisateur a la possibilité de ranger ses fichiers dans des sortes de classeurs (FOLDERS) pouvant eux-mêmes contenir d'autres classeurs, etc...
Le fichier sera identifié par son nom plus extension, et par le chemin qu'il faut suivre à travers les différents classeurs pour l'atteindre.
Exemple :
A :/maison/bureau/tiroir/déclaration



REMARQUE : un programme écrit en C doit obligatoirement se terminer par l'extension « c » → xxxxx.c.

Accessoires :
Les accessoires (DESK.ACC) sont des petits programmes utilitaires, disponibles aussitôt après le chargement du TOS et que l'on peut sélectionner à partir de la souris.
Deux de ces accessoires sont particulièrement utiles dans une optique de développement :
a - FILE - qui permet de visualiser des fichiers sur disquette ou dans des FOLDERS, d'ouvrir ou de fermer des FOLDERS, de changer de nom de fichiers/FOLDERS ou encore de lancer l'exécution de programmes sous GEM.
b - OPTIONS - qui permet entre autres de modifier la façon dont un programme va s'exécuter : option « Install Application ». En effet pour lancer l'exécution d'un programme sous GEM, il suffit en général de pointer la souris sur l'icone correspondant et de clicker deux fois ; le système charge alors le programme de la disquette, dans la mémoire centrale et donne la main au programme pour qu'il s'exécute. Cependant dans certains cas il faut donner au programme des paramètres (paramètres d'entrée) nécessaires au fonctionnement correct du programme.
Comment faire dans ce cas !!
Il suffit de sélectionner dans « Install Application » le rectangle « TOS takes parameters ».
Si on clicke deux fois sur l'icone représentant le programme GEM, avant de lancer l'exécution de programme, va afficher une grille à l'écran dans laquelle on pourra saisir les paramètres d'entrée du programme.
Une fois la grille validée (clicker sur OK), GEM transmet les paramètres au programme et lance l'exécution.
REMARQUE : les programmes avec une extension « TTP » disposent généralement directement de cette faculté.

3 - ENSEMBLE DE DEVELOPPEMENT (TOOLKIT).

Pour pouvoir utiliser un langage comme « C », il faut mettre en œuvre un certain nombre d'outils logiciels :
— programmes utilitaires
— fichiers de commandes
— bibliothèques de fichiers standarts utilisés au moment de la compilation et de l'édition de liens.
Les descriptions qui vont suivre s'inspirent spécialement de deux ensembles de développement disponibles à l'heure actuelle sur l'ATARI :
— le « PROGRAMMER'S TOOLKIT » de Digital Research
« HIPPO-C » de Hippopotamus Software.

Programmes utilitaires (commandes)

1. Interpréteur de commandes :
Si la présentation fenêtres icones est particulièrement commode pour l'utilisation courante de programmes ou pour la consultation de fichiers, elle s'avère en revanche peu pratique pour le développement de programmes, surtout en ce qui concerne la saisie directe de commandes au clavier, d'ou l'intérêt d'un programme capable de court circuiter la présentation de GEM pour la saisie des commandes, tout en maintenant la possibilité d'accéder aux commandes courantes du « bureau ». C'est ce que l'on appelle un interpréteur de commandes. C'est le premier programme a « clicker » s'il fait partie du TOOLKIT.
Programmes utilitaires généralement disponibles sous un interpréteur de commandes :



EVELOPPER SUR L'ATARI 520 ST

— création/visualisation/impression de fichiers.
— listage des fichiers et FOLDERS contenus sur les disquettes.
— copie/renommage/destruction de fichiers
— éditeur de texte (pour écrire ses programmes)
— compilateur/assembleur/éditeur de liens (pour produire les programmes).
— commande pour redonner la main au GEM et à la souris.

Toolkit DR : disponible, mais pas vraiment utilisable pour toutes les commandes.
HIPPO-C : HOS.PRG → premier programme à « clicker ».

2. fichier de commandes :
Il est souvent commode de regrouper des commandes qui portent sur un même fichier dans ce qu'on appelle un fichier de commande.
Exemple :
cc.prg est un programme qui transforme un fichier source écrit en « C » en un source « assembleur »
as.prg est un programme qui transforme un fichier source assembleur en un fichier objet.
On souhaite compiler (passage de source C à objet) un fichier toto.c qui se trouve dans le FOLDER « travail » sur la disquette B :
première méthode (lourde) :
Après avoir attribué l'option « TOS takes parameters » aux programmes cc.prg et as.prg ;
— « clicke cc.prg » → grille de saisie
— saisie du nom du fichier : B :/travail/toto.c
— Attente fin compilation
« clicke as.prg » → grille de saisie
— saisie du nom de fichier assembleur : B :/travail/toto.s
— Attente fin assemblage
— éventuellement suppression du fichier toto.s une fois que l'on a obtenu le fichier objet.
deuxième méthode (avec fichier de commande) :
saisie sous éditeur de texte des lignes suivantes :
cc.prg $1c. → produit un fichier $1.s
as.prg $1.s → produit un fichier $1.o
rm.prg $1.s → efface le fichier $1.s qui ne sert plus à rien. On sort de l'éditeur en donnant l'extension « bat » au fichier batch (fichier de commandes) ainsi constitué, afin de préciser sa nature à l'interpréteur de commandes.
en donnant au fichier le nom compilbat on peut utiliser compil en tant que commande :
compil B :/travail/toto
ce qui aura pour effet de lancer successivement toutes la commandes saisies dans compil.bat, $1 étant remplacé par B :/travail/toto.
$1 est une variable du fichier de commande. Un fichier de commande dispose généralement d'une dizaine de variables $1, $2, $n..., qui prendront les valeurs des paramètres d'entrée du fichier de commandes en fonction de l'ordre dans lequel ils auront été saisis.
Le Toolkit de DR fournit un programme spécialement destiné au lancement de fichier de commandes : BATCH.TTP.
pour l'utiliser, il faut clicker, saisir compil toto, et c'est parti.

REMARQUE :

les paramètres correspondant aux commandes de compilation et d'assemblage sont volontairement simplifiés pour la clarte de l'exposé (pour autant que quelque chose soit clair là-dedans).

3. Mise en œuvre d'un compilateur :
— D'abord saisie au moyen d'un éditeur de texte ed.prg pour HIPPO C, mince.ttp pour le TOOLKIT du programme suivant : toto.c.

```
ligne 1   # include <stdio.h>
ligne 2   main ()
ligne 3   {
ligne 4   printf (« hello Dolly/n ») ;
ligne 5   } ;
```

la première ligne du programme sera traitée par le compilateur.
ligne 1 # include <stdio.h>
Au moment de la compilation, le contenu du fichier source stdio.h est rajoute a toto.c, puis une deuxième passe de compilation s'exécute sur l'ensemble des fichiers « includes » (inclus en bon français) et le texte contenu dans toto.c.
A quoi cela sert-il ?
Essentiellement à attribuer des noms plus parlant à un certain nombre de constantes, tant systèmes, que personnelles, par exemple on peut créer un fichier couleur.h avec # définе ROUGE 1
puis écrire le programme suivant :

```
ligne 1   # include « couleur.h »
ligne 2   main ()
ligne 3   {
          int coul ;
ligne 4   coul = ROUGE ;
ligne 5   } ;
```

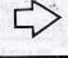
Dans ce programme, coul va prendre la valeur 1.
— compilation du programme par une commande du type c toto.
En fait « c » est un fichier de commande qui, dans une première étape, va transformer le source de départ en source assembleur toto.s, toto.s étant à son tour assemblé par un programme as.prg également contenu dans le fichier de commande.
Le résultat final de la compilation toto.o n'est ni un fichier source, ni un fichier binaire directement exécutable, c'est simplement le résultat de la commande d'assemblage contenue dans le fichier de commande.
Par ailleurs toto.c fait appel à une fonction « printf » qui n'est pas définie dans le fichier toto.c. On peut donc supposer qu'il manque encore une étape dans laquelle Il faudra rajouter aux xxx.o de notre programme les xxx.o standart fournis avec le kit de développement et qui contiennent les fonctions standarts C, systèmes et graphiques appliquables sur l'ATARI.
Généralement ces fichiers.o sont disposés dans un FOLDER nomme « LIB »
L'opération qui consiste à regrouper tous les xxxxx.o en un fichier exécutable s'appelle une édition de liens.
En résumé l'édition de liens.
→ réunit tous les xxxx.o donnés en paramètre de la commande avec les xxxx.o de la bibliothèque,
→ transforme le résultat en un fichier binaire qui contient toutes les instructions directement exécutables sur la machine,
→ établit l'adresse de chargement du programme qu'il place en tête fichier résultat.

4 - CONSEILS PRATIQUES

Les différents fichiers énumérés précédemment prennent beaucoup de place, et il est quasiment impossible de les faire tenir sur une disquette de 360 K.
Avec un seul lecteur, il vaut mieux adopter la répartition suivante :
1) éditeur de texte → disquette A
2) fichier.h → disquette B
+ fichier de commandes de compilation
+ programmes appelés dans fichiers « batch »
3) bibliothèque.o
→ disquette C
+ fichier de commandes d'édition de liens
+ programmes appelés dans fichiers « batch »,

Utilisation :
— saisie du programme sur la disquette A
— recopier programme sur la disquette B
— compiler sur la disquette B → « nom fichier ».o
— recopier « nom fichier ».o sur la disquette C
— éditer de liens sur C → « nom fichier ».prg
— lancer l'exécution du programme sous GEM ou sous le contrôle de l'interpréteur.

Si vous disposez d'un lecteur de disquette de capacité plus importante, vous pouvez essayer de regrouper A et B sur une même disquette.
Avec deux lecteurs, dont un de 720K, le mieux est de réunir A,B,C sur une disquette de 720K, le programme source et les fichiers résultants étant regroupés sur le deuxième lecteur.
Pour terminer voici des exemples de fichiers BATCH à utiliser avec une configuration de type CB pour compiler et éditer des liens sans trop de problèmes.



Pour finir voici un exemple type de programme utilisant GEM
:
```
/*----------------- DEBUT DU PROGRAMME
-------------------------*/

/* INCLUDE */
/*----------*/
£include <define.h>     /* definitions des variables
necessaires      */
£include <gemdefs.h>    /* a l|acces aux routines graphiqhes
*/
£include <obdefs.h>     /* du GEM
*/
£include <vdibind.h>
£include <osbind.h>     /* Fonctions systemes gemdos() et
appels      */
                        /* CP/M 68K .
*/

/* Definitions des variables utilisees par GEM :
*/
/* TOUJOURS LES DECLARER .......
*/
int     control 12 ;
int     intin 128 ;
int     ptsin 128 ;
int     intout 128 ;
int     ptsout 128 ;

int     work_in 11 ;
int     work_out 57 ;

extern int gl_apid;

int     handle;     /* Identificateur de votre ecran logique
*/
int     phys_handle;/* Identificateur de votre ecran physique
*/

int     gl_wchar;
int     gl_hchar;
int     gl_wbox;
int     gl_hbox;

main()          /* corps principal du programme.ca commence
ici */

  /* appel fonction locale debut_prog()  */
  /*-----------------------------------  */
    debut_prog();

  /* appel de votre programme          */
  /*---------------------------        */
    votre_prog();

  /* appel fin programme / fin_prog()  */
  /*-----------------------------------*/
    fin_prog();

  /* Fin du programme principal        */

/* Definitions des fonctions appelees dans le main / */
debut_prog()

int     i;

appl_init();    /* SURTOUT NE PAS OUBLIER ....       */
phys_handle = graf_handle( &gl_wchar , &gl_hchar ,
                           &gl_wbox  , &gl_hbox   );

for (i = 0 ; i < 10 ; work_in i++  = 1 )
    work_in 10  = 2;

handle = phys_handle:
v_opnwk(work_in , &handle , work_out);
   /* --------- fin debut_prog() --------- */

/*-----------------------------*/
fin_prog()

v_clswk( handle ); /* Fermeture de la station virtuelle */
appl_exit();       /* Fin de l|application              */

/*-----------------------------*/
votre_prog()

...
...
...
   vous ecriver votre programme ici
...
...
...
```

Je vous conseille vivement d|utiliser les fonctions
debut_prog()
et fin_prog() dans vos programmes , si vous compter utiliser

les fonctions du type VDI ou AES de GEM .

Voila , c|est fini .....



COMMENT GERER GEM VDI ET GEM AES EN BASIC

ST MAG — BASIC

On a raconté un peu tout et n'importe quoi sur le ST-Basic. Il est vrai que l'on était en droit d'attendre un superbe basic structuré avec procédures et structures de blocs. Mais ce n'est pas une raison suffisante pour le jeter. Il est assez rapide et permet de réaliser nombres d'applications interessantes. Je vous propose ici, d'apprendre à gerer GEM. Cet apprentissage sera d'autant plus utile que la gestion en Basic du VDI et de l'AES est assez semblable en C ou en langage Assembleur 68000.

Petit Rappel: Qu'est-ce que la VDI et l'AES.
La VIRTUAL DEVICE INTERFACE (VDI) est une partie de GEM qui contient toutes les routines de dessin, de couleurs, de remplissage, et d'impression.
L'AES contient, quant à lui les routines de gestion des fenêtres, de la souris, des messages d'alerte.
Il existe en ST-BASIC deux instructions particulières permettant d'accéder aux routines de GEM: GEMSYS et VDISYS.
GEMSYS donne accès à l'AES, et VDISYS à toutes les fonctions du VDI (sauf quatres spéciales).

Les variables globales:
GEM utilise pour transmettre les paramètres aux fonctions, 5 tableaux dont les noms sont imposés. On les utilise pour appeler les fonctions du VDI:
-CONTRL: Il est utilisé pour passer des paramètres en entrée. Son premier élément contient toujours le code référence (ou code opération)de la fonction appelée. Le 3ème élément (CONTRL+6) contient le nombre de paramètres transmis au tableau INTIN. Le 4ème (CONTRL+8) indique le nombre de valeurs renvoyées par la fonction aprés son exécution. Ces valeurs sont contenues dans le tableau INTOUT.
-Les tableaux PTSIN et PTSOUT contiennent les coordonnées sur l'écran des points utilisés par les fonctions graphiques appelées.

Pour les fonctions de l'AES on utilise la structure GB et la fonction GEMSYS(code opération fonction AES).

Commentaires des exemples:

Les exemples ont été conçus pour être exécutés sur un moniteur Monochrome. Cependant ils marchent aussi en couleur.

1) RECTANGLE: Cette fonction est étrangement absente du Basic. Elle dessine un rectangle rempli. La couleur et le style de remplissage sont définis par l'instruction BASIC "COLOR". PTSIN contient l'abscisse du coin supérieur gauche, et PTSIN+2 l'ordonnée. PTSIN+4 contient l'abscisse du coin inférieur droit, et PTSIN+6 l'ordonnée.

2) RECTANGLE ARRONDI: Cette fonction déssine un RECTANGLE AUX COINS ARRONDIS

3) RECTANGLE ARRONDI REMPLI: idem 1)

4) TEXTE: Cette fonction fait un peu la même chose que GOTOXY sauf que cette fois ci ce ne sont pas des colonnes ou des lignes de caractères qui sont désignées mais des coordonnées en PIXELS. Ceci permet une impression plus précise mais aussi plus rapide des textes.
CONTRL+6 contient le nombre de lettres à imprimer sur l'écran.
INTIN contient chaque caractère exprimé sur 16 bits.
PTSIN l'abscisse du premier caractère.
PTSIN+2 l'ordonnée de ce caractère.

5) LARGEUR DE LIGNE: Encore une fonction graphique absente dans le BASIC. La largeur de la ligne est entrée dans PTSIN.

6) FIN DE LIGNE: On peut changer la forme des bouts d'une droite.
INTIN contient le STYLE du début de doite.
INTIN+2 celui de la fin de la ligne.
Il y a trois valeurs possibles: 0 Forme angulaire (prise par défaut).
1 Flèche.
2 Forme arrondie.

7) TAILLE DES CARACTERES: La hauteur des caractères est entrée dans PTSIN+2. La valeur d'origine est 13 en monochrome, et 7 en résolution moyenne.

8) STYLE des caractères:

bit 0 ->	valeur=1	------------Caractères gras
1	2	fins
2	4	italiques
3	8	soulignés
4	16	éfilés
5	32	avec un ombre

On peut ajouter 2 valeurs pour mixer les différents styles.

9) ROTATION: Voici une fonction trés utile pour faire des présentations originales de vos programmes. Entrez dans INTIN l'angle désiré (0,900,1800,3600).

10) MONTRER OU CACHER LA SOURIS: Il peut parfois être utile de cacher la souris pendant le déroulement d'un programme. Pour cela il faut utiliser la fonction HIDE MOUSE (Opcode=123). Pour rendre visible la souris il faut utiliser SHOW MOUSE (Opcode=122).

11) GESTION DE LA SOURIS SOUS L'AES: Cet exemple montre comment obtenir les coordonnées de la souris, et la touche appuyée. On peut aussi utiliser la fonction du VDI qui est développée dans le dernier exemple.

12) RECTANGLE QUI GRANDIT PUIS DIMINUE: GEMSYS(73) fait grandir le rectangle et GEMSYS(74) fait éxactement l'inverse.

13) FORME DE LA SOURIS: Il existe plusieurs formes de curseur graphique pré-enregistrées. Vous pouvez grâce à cette fonction choisir le dessin que vous voulez utiliser dans votre programme. Pour cela il faut entrer le code du dessin voulu dans GINTIN puis appeler GEMSYS(78).
0=FLECHE NORMALE, 1=CURSEUR, 2=ABEILLE, 3=DOIGT TENDU, 4=MAIN OUVERTE, 5-6 ET 7 SONT DES CROIX.
Pour voir la forme de la souris changer je vous conseille de bouger la souris pendant l'exécution du programme.

14) DEPLACEMENT D'UN RECTANGLE: Cette fonction peut être utilisée conjointement avec celles de l'exemple 12 pour créer des effets optiques amusants.

15) GESTION DE LA SOURIS SOUS VDI: Cette fonction est plus simple et plus rapide que celle de l'AES (exemple 11).

Voilà, je m'arrête là. Je pense que ces quelques exemples sont suffisants pour comprendre comment gérer la VDI et l'AES en BASIC. Mais il y a bien d'autres fonctions disponibles. Celles du VDI sont quasiment toutes accessibles et en utilisant les exemples ci-dessus et le "LIVRE DU GEM" de Micro Application vous devriez facilement les retrouver. Celles de l'AES sont plus difficiles à atteindre. En fait seules les fonctions graphiques de l'AES sont facilement accessibles. En essayant d'utiliser les autres vous risquez fortement de "planter" le système...

```
10   REM Exemple 1: RECTANGLE
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2
40   FOR T=1 TO 16
50   COLOR 1,1,1,T,2:REM COULEUR DE REMPLISSAGE
60   POKE CONTRL,114:REM OPCODE DE "FILL RECTANGLE"
70   POKE CONTRL+2,2
80   POKE CONTRL+6,0
90   POKE PTSIN,50+T*10:REM COORDONNEE EN X DU 1ER POINT
100  POKE PTSIN+2,T*10+20:REM COORDONNES EN Y DU 1ER POINT
110  POKE PTSIN+4,250+T*10:REM COORDONNEES EN X DU PT DIAGONALEMENT OPPOSE
120  POKE PTSIN+6,200+10*T:REM COORDONNEES EN Y
130  VDISYS(1)
140  NEXT T
```

```
10   REM Exemple 2: RECTANGLE AUX COINS ARRONDIS
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   POKE CONTRL,11:REM OPCODE DE "ROUNDED RECTANGLE"
50   POKE CONTRL+2,2
60   POKE CONTRL+6,0
70   POKE CONTRL+10,8:REM IDENTIFICATEUR DE LA FONCTION
80   POKE PTSIN,50:REM COORDONNEE EN X DU 1ER POINT
90   POKE PTSIN+2,50:REM COORDONNES EN Y DU 1ER POINT
100  POKE PTSIN+4,250:REM COORDONNEES EN X DU PT DIAGONALEMENT OPPOSE
110  POKE PTSIN+6,200:REM COORDONNEES EN Y
120  VDISYS(1)
```

```
10   REM Exemple 3: RECTANGLE REMPLI AUX COINS ARRONDIS
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2
35   FOR T=1 TO 8:COLOR 1,1,1,T*2,2
40   POKE CONTRL,11:REM OPCODE DES "FONCTIONS GRAPHIQUES DE BASE"
50   POKE CONTRL+2,2
60   POKE CONTRL+6,0
70   POKE CONTRL+10,9:REM IDENTIFICATEUR DE LA FONCTION FILLED ROUNDED
RECTANGLE
80   POKE PTSIN,50+T*10:REM COORDONNEE EN X DU 1ER POINT
90   POKE PTSIN+2,50+T*10:REM COORDONNES EN Y DU 1ER POINT
100  POKE PTSIN+4,250-T*10:REM COORDONNEES EN X DU PT DIAGONALEMENT OPPOSE
110  POKE PTSIN+6,200-T*10:REM COORDONNEES EN Y
120  VDISYS(1)
130  NEXT T
```

```
10   REM Exemple 4: TEXTE au pixel prs
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   A$=" ST MAGASINE"
50   POKE CONTRL,8:REM OPCODE DE "TEXT"
60   POKE CONTRL+2,1
70   POKE CONTRL+6,LEN(A$):REM LONGUEUR CHAINE
80   FOR T=0 TO LEN(A$)-1
90   POKE INTIN+2*T,ASC(MID$(A$,T+1,1)):REM CHAINE ENTREE EN MEMOIRE
100  NEXT T
110  Y=52
120  FOR X=0 TO 520 STEP 3
130  POKE PTSIN,X
140  POKE PTSIN+2,Y
150  VDISYS(1)
160  Y=Y+1
170  NEXT X
```

```
10   REM Exemple 5: LARGEUR DE LIGNE
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   POKE CONTRL,16:REM OPCODE DE "POLYLINE WIDTH"
50   POKE CONTRL+2,1
60   POKE CONTRL+6,0
70   Y=2
80   FOR T=1 TO 15
90   COLOR 1,1,T
100  POKE PTSIN,T:REM LARGEUR DE LIGNE
110  POKE PTSIN+2,0
120  VDISYS(1)
130  LINEF 50,Y,300,Y
140  Y=Y+20
150  NEXT T
```

```
10   REM Exemple 6: STYLE FIN DE LIGNE
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   POKE CONTRL,108:REM OPCODE DE "POLYLINE END STYLES"
50   POKE CONTRL+2,0
60   POKE CONTRL+6,2
70   Y=10
```

```
80   FOR T=0 TO 2
90   COLOR 1,1,T+1
100  POKE INTIN,T:REM FORME DEBUT DE LIGNE
110  POKE INTIN+2,T:REM FORME FIN DE LIGNE
120  VDISYS(1)
130  LINEF 50,Y,300,Y
140  Y=Y+40
150  NEXT T
```

```
10   REM Exemple 7: TAILLE DES CARACTERES
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   POKE CONTRL,12:REM OPCODE DE "CHARACTER HEIGHT"
50   POKE CONTRL+2,1
60   POKE CONTRL+6,0
70   POKE PTSIN,0
80   FOR T=1 TO 30
90   POKE PTSIN+2,T:REM HAUTEUR DES CARACTERES
100  VDISYS(1)
110  GOTOXY 5,1:PRINT "S"
120  GOTOXY 5,3:PRINT "T"
130  GOTOXY 5,6:PRINT "M"
140  GOTOXY 5,8:PRINT "A"
150  GOTOXY 5,10:PRINT "G"
160  NEXT T
170  POKE PTSIN+2,13:VDISYS(1):REM RETOUR AU CARACTERES NORMAUX
```

```
10   REM Exemple 8: STYLE DES CARACTERES
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   POKE CONTRL,106:REM OPCODE DE "CHARACTER SPECIAL EFFECTS"
50   POKE CONTRL+2,0
60   POKE CONTRL+6,1
70   FOR T=0 TO 5
80   POKE INTIN,(2^T):REM STYLE DES CARACTERES
90   VDISYS(1)
100  GOTOXY 5,5:PRINT "S T   M A G A S I N E "
105  FOR P=1 TO 300:NEXT P:REM BOUCLE DE RALENTISSEMENT
110  NEXT T
120  POKE INTIN,0:VDISYS(1):REM Retour  la normale
```

```
10   REM EXEMPLE 9: ROTATION DE CARACTERES
20   REM c)1986 ST MAGASINE
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   POKE CONTRL,13:REM OPCODE DE "CHARACTER BASELINE VECTOR"
50   POKE CONTRL+2,0
60   POKE CONTRL+6,1
70   FOR P=1 TO 3
80   FOR T=0 TO 3500 STEP 900
90   POKE INTIN,T:REM ON FAIT TOURNER LES CARACTERES DE T DEGRES
100  VDISYS(1)
110  GOTOXY 8,5:PRINT "S"
120  GOTOXY 9,5:PRINT "T"
130  GOTOXY 11,5:PRINT "M"
140  GOTOXY 12,5:PRINT "A"
150  GOTOXY 13,5:PRINT "G"
160  CLEARW 2
170  NEXT T,P
180  POKE INTIN,0:VDISYS(1)
```

```
10   REM EXEMPLE 10:MONTRER ET CACHER LA SOURIS->ICI ON LA FAIT CLIGNOTER
20   REM C)1986 ST MAGASINE
30   FOR T=1 TO 200
40   POKE CONTRL,122:REM OPCODE DE SHOW MOUSE
50   POKE CONTRL+2,0
60   POKE CONTRL+4,1
70   POKE INTIN,0
80   VDISYS(1):REM ON MONTRE LA SOURIS
90   POKE CONTRL,123:REM OPCODE DE HIDE MOUSE
100  POKE CONTRL+2,0
110  POKE CONTRL+4,0
120  VDISYS(1):REM ON CACHE LA SOURIS
130  NEXT T
140  POKE CONTRL,122
150  POKE CONTRL+2,0
160  POKE CONTRL+4,1
170  VDISYS(1):REM RETOUR A L¤ETAT NORMAL
```

```
10   REM EXEMPLE 11: GESTION DE LA SOURIS
20   REM C)1986 ST MAGASINE
30   BS=0:FULLW 2:CLEARW 2:COLOR 1,1,1
35   GOTOXY 0,1:PRINT" APPUYEZ SUR LES 2 BOUTONS A LA FOIS POUR SORTIR."
40   WHILE(BS<3)
50   GOTOXY 2,2:PRINT XS;" Abscisse de la souris.  "
60   GOTOXY 2,3:PRINT YS;" Ordonne de la souris.  "
70   GOTOXY 2,4:PRINT BS;" Etat du bouton de la souris."
80   GOSUB SOURIS
90   WEND
100  END
9000 SOURIS:
9010 A£=GB
9020 GRINOUT=PEEK(A£+12)
9030 GEMSYS(79):REM CODE AES DE GRAF_MKSTATE.
9040 XS=PEEK(GRINOUT+2)
9050 YS=PEEK(GRINOUT+4)
9060 BS=PEEK(GRINOUT+6)
9070 RETURN
```

```
10   REM EXEMPLE 12: RECTANGLE GRANDISSANT ET EN RETRECISSANT
20   REM C)1986 ST MAGASINE.
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   A£=GB
50   CONTROL=PEEK(A£)
60   GINTIN=PEEK(A£+8)
70   POKE GINTIN,150:REM X DU PLUS PETIT RECTANGLE
80   POKE GINTIN+2,90:REM Y DU PLUS PETIT RECTANGLE
90   POKE GINTIN+4,20:REM LARGEUR DU PLUS PETIT RECTANGLE
100  POKE GINTIN+6,20:REM HAUTEUR DU PLUS PETIT RECTANGLE
110  POKE GINTIN+8,50:REM X DU PLUS GRAND RECTANGLE
120  POKE GINTIN+10,50:REM Y DU GRAND RECTANGLE
130  POKE GINTIN+12,200:REM LARGEUR DU PLUS GRAND RECTANGLE
140  POKE GINTIN+14,150:REM HAUTEUR DU PLUS GRAND RECTANGLE
150  GEMSYS(73):REM CODE AES DE GROWBOX(RECTANGLE GRANDISSANT)
160  LINEF 50,23,250,23:LINEF 250,23,250,173:LINEF 250,173,50,173
170  LINEF 50,173,50,23
180  GOTOXY 6,4:PRINT  "ST MAGASINE"
190  GOTOXY 6,7:PRINT  "AES ET VDI."
200  GOTOXY 0,0:INPUT "PRESSEZ ENTER ";A$
210  COLOR 1,1,0
220  LINEF 50,23,250,23:LINEF 250,23,250,173:LINEF 250,173,50,173
230  LINEF 50,173,50,23
240  GEMSYS(74):REM CODE AES DE SHRINKBOX(RECTANGLE DIMINUANT)
```

ST MAG

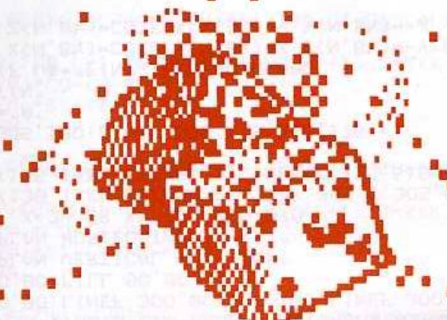
BASIC

```
10   REM EXEMPLE 13: FORME DE LA SOURIS
20   REM C)1986 ST MAGASINE.
30   A£=GB
40   CONTROL=PEEK(A£)
50   ADDRIN=PEEK(A£+16)
60   GINTIN=PEEK(A£+8)
70   POKE GINTIN,1:GOSUB PAUSE:REM CURSEUR
80   POKE GINTIN,2:GOSUB PAUSE:REM ABEILLE
90   POKE GINTIN,3:GOSUB PAUSE:REM DOIGT
100  POKE GINTIN,4:GOSUB PAUSE:REM MAIN
110  POKE GINTIN,5:GOSUB PAUSE:REM CROIX FINE
120  POKE GINTIN,6:GOSUB PAUSE:REM CROIX EPAISSE
130  POKE GINTIN,7:GOSUB PAUSE:REM CROIX ENTOURE
140  POKE GINTIN,0:REM RETOUR A LA FLECHE
150  GEMSYS(78)
160  END
170  PAUSE:
180  GEMSYS(78)
190  FOR T=1 TO 400:NEXT T
200  RETURN
```

```
10   REM EXEMPLE 14: RECTANGLE QUI BOUGE
20   REM C)1986 ST MAGASINE.
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   A£=GB
50   CONTROL=PEEK(A£)
60   GINTIN=PEEK(A£+8)
70   POKE GINTIN,200:REM LARGEUR DU RECTANGLE
80   POKE GINTIN+2,150:REM HAUTEUR DU RECTANGLE
90   POKE GINTIN+4,00:REM X DE DEPART
100  POKE GINTIN+6,20:REM Y DE DEPART
110  POKE GINTIN+8,300:REM X D©ARRIVEE
120  POKE GINTIN+10,100:REM Y D©ARRIVEE
130  FOR T= 1 TO 5
140  GEMSYS(72):REM CODE AES DE MOVEBOX
150  FOR P=1 TO 300:NEXT P
160  NEXT T
170  COLOR 1,1,0
180  LINEF 50,23,250,23:LINEF 250,23,250,173:LINEF 250,173,50,173
```

```
10   REM DERNIER EXEMPLE:GESTION DE LA SOURIS SOUS VDI.
20   REM C)1986 ST MAGASINE
30   FULLW 2:CLEARW 2:COLOR 1,1,1
40   WHILE BS<3
50   GOTOXY 5,5:PRINT XS;"  ";YS;"  ";BS
60   GOTOXY 5,5:PRINT "                  "
70   GOSUB MOUSE
80   WEND
90   END
9900 MOUSE:
9910 POKE CONTRL,124
9920 POKE CONTRL+2,0
9930 POKE CONTRL+4,0
9935 VDISYS(1)
9940 BS=PEEK (INTOUT)
9950 XS=PEEK (PTSOUT)
9960 YS=PEEK (PTSOUT+2)
9970 RETURN
```

3 D



Ce programme permet de dessiner des objets et de les représenter en perspective sur l'écran noir et blanc.

Un objet est constitué de modules manipulables séparément (rotation, translation, modification de sa taille).

Lorsque vous mettez le programme en route, apparait le MENU PRINCIPAL qui donne le choix entre créer un objet et en récupérer un d'une disquette (attention, il n'y a pas de catalogue, si l'objet n'existe pas, le programme se plante).

Dans les deux cas apparaît le même SOUS-MENU qui permet de créer un module, de le faire tourner, de le déplacer, de modifier sa taille ou encore de faire apparaître l'objet en son entier sur l'écran.

CREATION D'UN PROFIL: Chaque clic du bouton gauche de la souris crée un point à l'emplacement du curseur, deux points consécutifs sont reliés par une droite. Cliquer le bouton droit de la souris indique que la saisie est terminée. Le programme demande alors le nombre de faces à générer ainsi que l'angle (360° font un tour).

ROTATION: On vous demande l'axe de rotation (X,Y ou Z) puis l'angle de rotation.
X perpendiculaire au plan de l'écran.
Y horizontal.
Z vertical.

TRANSLATION: On vous demande les composantes du vecteur translation (X,Y,Z) (3 nombres séparés par une virgule).

GROSSISSEMENT: On vous demande un facteur.
facteur=1 pas de modification.
facteur>1 la taille est augmentée.
facteur<1 la taille est diminuée.

VISUALISATION: Après un moment d'attente apparait le dessin dans son ensemble, en cliquant le bouton gauche de la souris, apparait un menu vous permettant diverses manipulations sur le dessin dans son ensemble (rotations, zoom).

REMARQUES:

Toutes les sélections se font avec la souris (sauf une, je vous laisse la trouver).

Le programme vous avertit si vous commettez une erreur.

Le programme est modulaire et en général le listing est assez explicite, vous pouvez donc envisager des modifications. Je vous en suggère quelques unes

-Le s/s programme lecture écriture de fichiers n'est pas un modèle du genre (mais fonctionne).

-Dans la version actuelle, un objet est constitué de 3 modules de 400 points. Si cela ne correspond pas à vos besoins modifiez les lignes 50,60,70. (En particulier avec 500 Koctets vous risquez un out of memory).

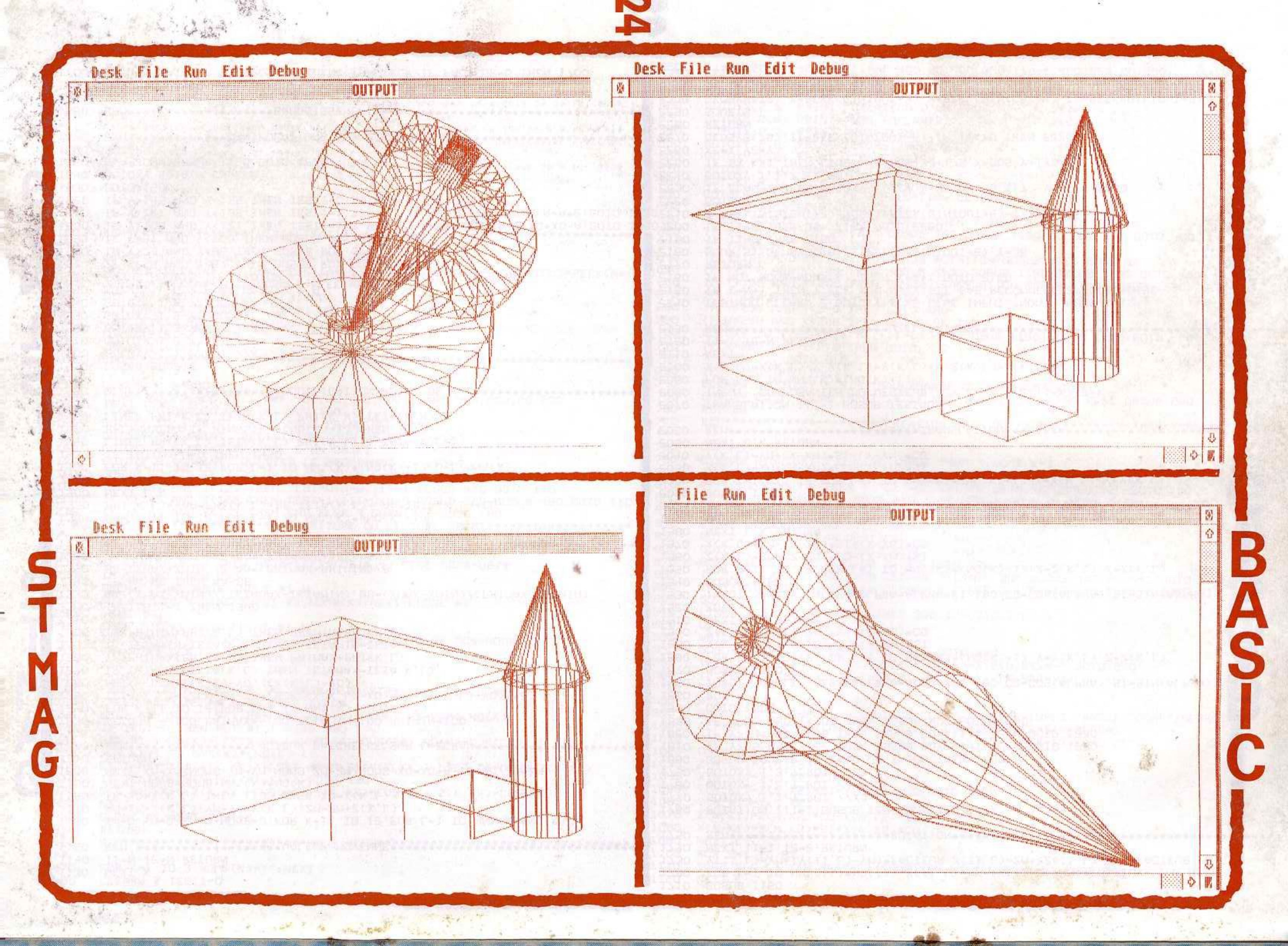
-Si vous voulez une perspective style "grand angle", diminuez la valeur de A en ligne 110.

Et maintenant à vous de jouer.

DROIN CHRISTIAN

```
10   REM***************INITIALISATION*************************************
20   A#=GB:XMIN=10:XMAX=610:YMIN=10:YMAX=340
30   CONTROL=PEEK(A#):GLOBAL=PEEK(A#+4):GINTIN=PEEK(A#+8):GINTOUT=PEEK(A#+12)
40   ADDRIN=PEEK(A#+16):ADDROUT=PEEK(A#+20)
50   DIM X(3,400),Y(3,400),Z(3,400)
60   DIM CHAINES$(3),NN%(3),O$(3),NO(3)
70   DIM TETA(3,400),PHI(3,400)
80   FULLW 2:PI=3.14159
90   COLOR 1,1,1,1,2
100  REM************************MENU PRINCIPAL*******************************
110  CLEARW 2:A=500
120  linef 20,20,400,20:linef 400,20,400,200:linef 400,200,20,200
130  linef 20,200,20,20:fill 100,100
140  GOTOXY 10,3:PRINT "**** MENU PRINCIPAL ****"
150  GOTOXY 2,5:PRINT "DEFINITION D'UN OBJET    :<1>"
160  GOTOXY 2,6:PRINT "RAPPEL D'UN OBJET        :<2>"
170  GOTOXY 2,7:PRINT "QUITTER LE PROGRAMME     :<3>"
180  GOSUB SOURIS:IF X<35 OR X>272 THEN ? CHR$(7):GOTO 180
190  IF Y>127 AND Y<139 THEN TEST=0:N=0:III=1:GOSUB 470:GOTO 110
200  IF Y>145 AND Y<156 THEN TEST=1:III=0:N=0:GOSUB 220:GOSUB 480:GOTO 110
210  IF Y>161 AND Y<173 THEN END ELSE ? CHR$(7);:GOTO 180
220  REM********************ECRITURE OU LECTURE OBJET*********************
230  GOTOXY 5,17:INPUT "NOM DE L'OBJET";OBJ$
240  IF LEN(OBJ$)>4 THEN ABJ$=LEFT$(OBJ$,4) ELSE ABJ$=OBJ$
250  IF TEST=0 THEN 290
260  OPEN "I",#1,ABJ$:INPUT #1,N
270  FOR K=1 TO N:LINE INPUT #1,NOM$(K):NEXT:CLOSE #1
280  GOTO 340
290  OPEN "O",#1,ABJ$:PRINT #1,N
300  FOR K=1 TO N:IF K=1 THEN CC$="X"+NOM$(K) ELSE CC$=NOM$(K)
310  PRINT #1,CC$
320  NEXT
330  CLOSE #1
340  FOR K=1 TO N:OPEN "R",#1,ABJ$+MID$(STR$(K),2,LEN(STR$(K))),12
350  FIELD #1,4 AS A$,4 AS B$,4 AS C$
360  LSET A$=O$(K):LSET B$=MKS$(NN%(K)):LSET C$=MKS$(NO(K))
370  IF TEST=0 THEN PUT #1,1:GOTO 390
380  GET #1,1:O$(K)=LEFT$(A$,1):NN%(K)=CVS(B$):NO(K)=CVS(C$)
390  FOR KK=1 TO NN%(K)*(NO(K)+1)
400  LSET A$=MKS$(X(K,KK)):LSET B$=MKS$(Y(K,KK)):LSET C$=MKS$(Z(K,KK))
410  IF TEST=0 THEN PUT #1,KK+1:GOTO 430
420  GET #1,KK+1:X(K,KK)=CVS(A$):Y(K,KK)=CVS(B$):Z(K,KK)=CVS(C$)
430  NEXT
440  CLOSE #1:NEXT:TEST=0
450  RETURN
460  REM*************************SOUS MENU**********************************
470  FOR K=1 TO 3:NOM$(K)="":NEXT
480  CLEARW 2:TEST1=0
490  linef 20,20,400,20:linef 400,20,400,250:linef 400,250,20,250
500  linef 20,250,20,20:fill 100,100
510  GOTOXY 10,3:PRINT "**** SOUS-MENU ****":PRINT
520  GOTOXY 2,5: PRINT "CREATION D'UN PROFIL          :<1>"
530  GOTOXY 2,6: PRINT "NON IMPLANTE                  :<2>"
540  GOTOXY 2,7: PRINT "ROTATION                      :<3>"
550  GOTOXY 2,8: PRINT "GROSSISSEMENT                 :<4>"
560  GOTOXY 2,9: PRINT "TRANSLATION                   :<5>"
570  GOTOXY 2,10:PRINT "NON IMPLANTE                  :<6>"
580  GOTOXY 2,11:PRINT "NON IMPLANTE                  :<7>"
590  GOTOXY 2,12:PRINT "VISUALISATION DESSIN          :<8>"
600  GOTOXY 2,13:PRINT "QUITTER                       :<9>"
610  GOSUB SOURIS:IF X<34 OR X>272 THEN ? CHR$(7);:GOTO 610
620  IF Y>127 AND Y<139 THEN GOSUB 780:GOSUB 1150:GOSUB 1230:GOTO 480
630  IF Y>145 AND Y<156 THEN GOTO 480
640  IF Y>162 AND Y<172 THEN GOSUB ROTATION:GOSUB 1230:GOTO 480
650  IF Y>178 AND Y<189 THEN GOSUB GROS:GOSUB 1150:GOSUB 1230:GOTO 480
660  IF Y>196 AND Y<208 THEN GOSUB TRANSLATION:GOSUB 1150:GOSUB 1230:GOTO 480
670  IF Y>213 AND Y<224 THEN GOTO 480
680  IF Y>230 AND Y<240 THEN GOTO 480
690  IF Y>248 AND Y<258 THEN 700 ELSE 710
700  GOTOXY 10,17:? CHR$(9);:I1=1:I2=N:GOSUB 1150:GOSUB 1230:GOTO 480
710  IF Y>260 AND Y<275 THEN GOTO 730
720  ? CHR$(7);:GOTO 610
730  IF III=0 THEN 760
740  CLEARW 2:GOTOXY 5,5:? "SAUVEGARDE DU FICHIER? (O/N)"
750  GOTOXY 20,17:LINE INPUT I$:IF NOT I$="O" THEN 760 ELSE GOSUB 220
760  RETURN
770  REM*********************CREATION PLAN OU PROFIL*************************
780  O$(N+1)="O"
790  CLEARW 2:GOTOXY 1,3:LINE INPUT "NOM ";NOM$:K=1
800  IF NOM$="FIN" THEN N=N-1:GOTO 1140
810  IF NOM$(K)=NOM$ THEN ? "CE NOM EXISTE DEJA:RECOMMENCEZ":GOTO 790
820  IF K<N-1 THEN K=K+1:GOTO 810 ELSE N=N+1:NOM$(N)=NOM$:III=1
830  LINEF 20,80,300,80:LINEF 300,80,300,130:LINEF 300,130,20,130
840  LINEF 20,130,20,80:FILL 50,90
850  GOTOXY 2,5:? "PLAN VERTICAL      <V>"
860  GOTOXY 2,6:? "PLAN HORIZONTAL    <H>"
870  GOSUB SOURIS:IF X<34 OR X>220 THEN 870
880  IF Y>127 AND Y<139 THEN CLEARW 2:LINEF 305,0,305,400:PLAN=1:GOTO 910
890  IF Y>145 AND Y<156 THEN CLEARW 2:LINEF 0,175,615,175:PLAN=2:GOTO 910
900  GOTO 870
910  LINEF 305,170,305,180:LINEF 300,175,310,175
920  COTE=0:B%=1:U$="A"
930  WHILE NOT U$="FIN"
940  GOSUB SOURIS1:IF U$="FIN" THEN GOTO 980
950  IF PLAN=1 THEN X(N,B%)=COTE:Y(N,B%)=X:Z(N,B%)=-Y:GOTO 970
960  IF PLAN=2 THEN Z(N,B%)=COTE:X(N,B%)=Y:Y(N,B%)=-X
970  B%=B%+1
980  WEND:B%=B%-1:NN%(N)=B%
990  GOTOXY 1,15:INPUT "NOMBRE DE COTES ";NO(N):INPUT "ANGLE TOTAL";ALPHAT
1000 ALPHA=PI*ALPHAT/(NO(N)*180):CO=COS(ALPHA):SI=SIN(ALPHA)
1010 FOR K=0 TO (NO(N)-1):FOR L=1 TO B%:K1=L+B%*(K+1):K2=L+B%*K
1020 ON PLAN GOTO 1030,1060
1030 Z(N,K1)=Z(N,K2)
1040 X(N,K1)=X(N,K2)*CO-Y(N,K2)*SI
1050 Y(N,K1)=X(N,K2)*SI+Y(N,K2)*CO:GOTO 1090
1060 Y(N,K1)=Y(N,K2)
1070 Z(N,K1)=Z(N,K2)*CO-X(N,K2)*SI
1080 X(N,K1)=Z(N,K2)*SI+X(N,K2)*CO
1090 NEXT L
1100 FOR L=1 TO B%
1110 SWAP X(N,L),X(N,B%+1-L):SWAP Y(N,L),Y(N,B%+1-L):SWAP Z(N,L),Z(N,B%+1-L)
1120 NEXT L
```

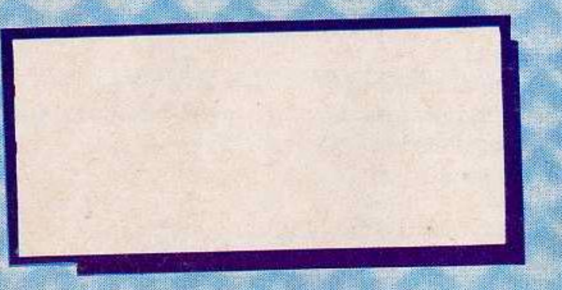
```
1130  NEXT K
1140  I1=N:I2=N:RETURN
1150  REM *******************VALEUR MOYENNE*********************************
*
1160  XM=0:YM=0:ZM=0:NOMB=0:FOR K=I1 TO I2:FOR L=1 TO NN%(K)*NO(K)
1170  XM=XM+X(K,L):YM=YM+Y(K,L):ZM=ZM+Z(K,L)
1180  IF XO>X(K,L) THEN 1190 ELSE XO=X(K,L)
1190  NEXT:NOMB=NOMB+NN%(K)*NO(K)
1200  NEXT:XM=XM/NOMB:YM=YM/NOMB:ZM=ZM/NOMB:XO=XO+A:YO=YM:ZO=ZM
1210  RETURN
1220  REM*****************FACTEUR D=ECHELLE SUR L=ECRAN**********************
*
1230  TETAMAX=-100:PHIMAX=-100:TETAMIN=100:PHIMIN=100
1240  FOR K=I1 TO I2:FOR L=1 TO NN%(K)*(NO(K)+1)
1250  TETA(K,L)=(Z(K,L)-ZO)/SQR((X(K,L)-XO)**2+(Y(K,L)-YO)**2)
1260  PHI(K,L)=(Y(K,L)-YO)/(X(K,L)-XO)
1270  IF TETAMAX<TETA(K,L) THEN TETAMAX=TETA(K,L)
1280  IF PHIMAX<PHI(K,L) THEN PHIMAX=PHI(K,L)
1290  IF TETAMIN>TETA(K,L) THEN TETAMIN=TETA(K,L)
1300  IF PHIMIN>PHI(K,L) THEN PHIMIN=PHI(K,L)
1310  NEXT L,K
1320  IF TEST1=1 THEN 1380
1330  AA=(YMAX-YMIN)/(TETAMAX-TETAMIN):BB=(XMAX-XMIN)/(PHIMAX-PHIMIN)
1340  IF AA>BB THEN AA=BB
1350. BB=(XMAX+XMIN)/2-AA*(PHIMAX+PHIMIN)/2
1360  CC=(YMAX+YMIN)/2+AA*(TETAMAX+TETAMIN)/2
1370  REM***********************DESSIN*************************************
*
1380  FOR K=I1 TO I2:FOR L=1 TO NN%(K)*(NO(K)+1)
1390  PHI(K,L)=+AA*PHI(K,L)+BB:TETA(K,L)=-AA*TETA(K,L)+CC
1400  NEXT L,K
1410  CLEARW 2
1420  FOR K=I1 TO I2:FOR L=1 TO NN%(K)*(NO(K)+1):U=L/NN%(K)
1430  IF INT(U)-U=0 THEN 1450
1440  LINEF PHI(K,L),TETA(K,L),PHI(K,L+1),TETA(K,L+1)
1450  H=L+NN%(K):IF H>NN%(K)*(NO(K)+1) THEN 1470
1460  LINEF PHI(K,L),TETA(K,L),PHI(K,H),TETA(K,H)
1470  NEXT L,K
1480  REM*******************MODIFICATION POINT DE VUE**********************
*
1490  GOSUB SOURIS
1500  GOTOXY 1,1:?"ROT //X"
1510  GOTOXY 1,2:?"ROT //Y"
1520  GOTOXY 1,3:?"ROT //Z"
1530  GOTOXY 1,4:?"ZOOM +"
1540  GOTOXY 1,5:?"ZOOM -"
1550  GOTOXY 1,6:?"FIN":GOTOXY 1,8
1560  GOSUB SOURIS:IF X<10 OR X>100 THEN ? CHR$(7):GOTO 1560
1570  IF Y>55 AND Y<67 THEN GOSUB ROX:GOTO 1230
1580  IF Y>73 AND Y<85 THEN GOSUB ROY:GOTO 1230
1590  IF Y>91 AND Y<103 THEN GOSUB ROZ:GOTO 1230
1600  IF Y>109 AND Y<121 THEN TEST1=1:?:? "  "+CHR$(9):A=A/2:XO=XO-A:GOTO 1230
1610  IF Y>127 AND Y<139 THEN TEST1=1:?:? "  "+CHR$(9):XO=XO+A:A=A*2:GOTO 1230
1620  IF Y>145 AND Y<156 THEN TEST1=0:RETURN
1630  GOTO 1560
1640  GOTOXY 4,15:? CHR$(9)
1650  A=A+ZOOM:GOSUB 1150:GOTO 1230
1660  RETURN
1670  REM*************MANIPULATIONS SUR DES MORCEAUX*************************
*
1680  REM*******************GROSSISSEMENT**********************************
*
1690  GROS:III=1:GOSUB TESTNOM:I1=K:I2=K
1700  INPUT "FACTEUR DE GROSSISSEMENT";FACTEUR:IF FACTEUR=0 THEN 1700
1710  GOSUB 1160
1720  FOR L=1 TO NN%(I1)*NO(I1):X(I1,L)=XM+(X(I1,L)-XM)*FACTEUR
1730  Y(I1,L)=YM+(Y(I1,L)-YM)*FACTEUR:Z(I1,L)=ZM+(Z(I1,L)-ZM)*FACTEUR
1740  NEXT:I1=1:I2=2:RETURN
1750  REM*********************ROTATION******************************************
*
1760  ROTATION:III=1:GOSUB TESTNOM:I1=K:I2=K
1770  GOTOXY 1,1:?"ROT //X"
1780  GOTOXY 1,2:?"ROT //Y"
1790  GOTOXY 1,3:?"ROT //Z"
1800  GOSUB SOURIS:IF X<10 OR X>100 THEN 1790
1810  IF Y>55 AND Y<67 THEN GOSUB ROX:I1=1:I2=N:GOTO 1230
1820  IF Y>73 AND Y<85 THEN GOSUB ROY:I1=1:I2=N:GOTO 1230
1830  IF Y>91 AND Y<103 THEN GOSUB ROZ:GOTO 1230
1840  GOTO 1800
1850  ROX:
1860  INPUT "ANGLE";ALPHA:ALPHA=ALPHA*PI/180:CO=COS(ALPHA):SI=SIN(ALPHA)
1870  GOSUB 1160
1880  FOR K=I1 TO I2:FOR L=1 TO NN%(K)*(NO(K)+1):Y=Y(K,L):Z=Z(K,L)
1890  Y(K,L)=YM+(Y-YM)*CO-(Z-ZM)*SI
1900  Z(K,L)=ZM+(Y-YM)*SI+(Z-ZM)*CO
1910  NEXT L,K:RETURN
1920  ROY:
1930  INPUT "ANGLE";ALPHA:ALPHA=ALPHA*PI/180:CO=COS(ALPHA):SI=SIN(ALPHA)
1940  GOSUB 1160
1950  FOR K=I1 TO I2:FOR L=1 TO NN%(K)*(NO(K)+1):Z=Z(K,L):X=X(K,L)
1960  Z(K,L)=ZM+(Z-ZM)*CO-(X-XM)*SI
1970  X(K,L)=XM+(Z-ZM)*SI+(X-XM)*CO
1980  NEXT L,K:RETURN
1990  ROZ:
2000  INPUT "ANGLE";ALPHA:ALPHA=ALPHA*PI/180:CO=COS(ALPHA):SI=SIN(ALPHA)
2010  GOSUB 1160
2020  FOR K=I1 TO I2:FOR L=1 TO NN%(K)*(NO(K)+1):X=X(K,L):Y=Y(K,L)
2030  X(K,L)=XM+(X-XM)*CO-(Y-YM)*SI
2040  Y(K,L)=YM+(X-XM)*SI+(Y-YM)*CO
2050  NEXT L,K:RETURN
2060  REM************************TRANSLATION**********************************
*
2070  TRANSLATION:III=1:GOSUB TESTNOM
2080  INPUT "COMPOSANTES DU VECTEUR:X,Y,Z";X,Y,Z:I1=K:I2=K
2090  FOR L=1 TO NN%(K)*(NO(K)+1)
2100  X(K,L)=X(K,L)+X:Y(K,L)=Y(K,L)+Y:Z(K,L)=Z(K,L)+Z
2110  NEXT
2120  I1=1:I2=N:RETURN
2130  REM*********************UTILITAIRES*************************************
*
2140  TESTNOM:CLEARW 2:GOTOXY 1,15:LINE INPUT "NOM ",NOM$:K=1
2150  IF K=N+1 THEN ? "CE NOM N=EXISTE PAS:RECOMMENCEZ":GOTO 2140
2160  IF NOT NOM$=NOM$(K) THEN K=K+1:GOTO 2150
2170  RETURN
2180  SOURIS1:GEMSYS(79):T1=PEEK(GINTOUT+6):T=38
2190  IF T1=0 THEN 2180
2200  IF T1=2 THEN U$="FIN":GOTO 2280
2210  X=PEEK(GINTOUT+2)-305:Y=PEEK(GINTOUT+4)-T-175
2220  GOTOXY 1,1:? "          ";
2230  IF PLAN=1 THEN GOTOXY 1,1:? X:GOTOXY 3,1:? -Y:GOTO 2250
2240  GOTOXY 1,1:? -X:GOTOXY 3,1:? Y
2250  IF B% <>1 THEN LINEF X2+305,Y2+175,X+305,Y+175
2260  X2=X:Y2=Y
2270  GEMSYS(79):T1=PEEK(GINTOUT+6):IF T1<>0 THEN 2270
2280  RETURN
2290  SOURIS:
2300  GEMSYS(79):X=PEEK(GINTOUT+2):Y=PEEK(GINTOUT+4):T1=PEEK(GINTOUT+6)
2310  IF T1<>1 THEN GOTO 2300
2320  RETURN
```


Desk File Run Edit Debug
OUTPUT
Desk File Run Edit Debug
OUTPUT
Desk File Run Edit Debug
OUTPUT
File Run Edit Debug
OUTPUT
ST MAG
BASIC

CATALOGUE des PRODUITS de la GAMME ATARI ST

printemps 86





Revendeurs:contactez-nous!



LITTERATURE

Les revues

ST MAGAZINE N°1 — **Pressimage**
Pourquoi acheter un ST - PCW Show - Les 150 premiers logiciels - Logo - Langage C - GEM j'aime - Communiquer avec son ST - Gestion: horizontale ou verticale - L'interface MIDI -

ST MAGAZINE N°2 — **Pressimage**
Le graphisme du ST - COMDEX Las Vegas - ST Index - LOGO, çà se corse - Demarrer en 68000 - Langage C: Accès à GEM - Le 68901 - Survol du Basic - GEM (2) - Developper sur ST -

ST MAGAZINE N°3 — **Pressimage**
(A paraitre Mars/Avril 86)

ST MAGAZINE N°4 — **Pressimage**
(A paraitre Juin/Juillet 86)

Les livres

LE NOUVEL ATARI ST — **Micro-Application**
Livre d'approche du ST. Assez ancien, il n'est à conseiller qu'à ceux qui se posent la question de l'achat d'un ST.

LE LIVRE DU LANGAGE MACHINE SUR ST
Livre d'approche du langage machine sur ST. Contient très peu d'éléments sur le ST, bon bouquin d'initiation néanmoins.

LE LIVRE DU GEM SUR ST — **Micro-Application**
Excellent ouvrage pour les développeurs sur ST. GEM et ses rapports avec les langages de programmation en particulier C et Assembleur. Indispensable !

LA BIBLE DU ST — **Micro-Application**
68000 - Processeurs spéciaux - Controleur 1722 - 68901 - 6850 - YM 2149 - Entrées/Sorties - Clavier - Vidéo - Interfaces MIDI, série, parrallèle - Ports cartouche, floppy, DMA - GEMDOS - BIOS - XBIOS - BIOS listing

TRUCS ET ASTUCES SUR ST — **Micro-Application**
Basic et GEM - Disque virtuel - Copie d'écran monochrome et couleur - Applications GEM - Nombreux listings

USING LOGO ON ATARI ST — **Glentop**
Ouvrage en anglais, mais essentiel pour ceux qui veulent utiliser pleinement la richesse du LOGO fourni avec toutes les machines.

ST' PROGRAMMER'S GUIDE — **Compute**
Orienté vers les débutants, une bonne initiation à la programmation sur ST.

THE ELEMENTARY ATARI ST — **Compute**
Introduction au ST. Chapitres sur le LOGO et le FORTH.

LES OUTILS DE PROGRAMMATION

Les systèmes d'exploitation

La machine est livrée avec le TOS (variante de CP/M 68K) et GEM
Un boîtier d'extension permettant de faire tourner MS DOS et donc les logiciels de l'IBM PC sera disponible avant Septembre 86.
Des rumeurs de plus en plus insistantes font état de la disponibilité d'UNIX sur le ST fin 86/ début 87.
En outre un émulateur APPLE II sera disponible au cours de l'année.

CP/M 80 version 2.2 — **Atari**
Cet émulateur permet de faire tourner des logiciels sous CP/M tel que Wordstar, Multiplan ou Turbo Pascal.

BOS — **Bos**
A la fois langage et système d'exploitation, BOS est disponible sur le ST et permet de faire tourner un grand nombre de logiciels professionnels d'origine anglaise.

MEMDOS — **Memsoft**
Memdos est l'équivalent français de Bos. Implémenté sur ST, il va permettre de disposer rapidement d'un grand nombre de logiciels professionnels horizontaux et verticaux.

Les langages

La machine est livrée avec un bon LOGO et un BASIC plus discutable. Tous les grands langages sont ou vont être disponibles dans les trois prochains mois.

HENRY'S BASIC — **Philon**
Interpréteur BASIC orienté vers les débutants.

COMPILATEUR BASIC — **Philon**
Pour tous ceux qui souhaitent développer en BASIC sur ST. On écrit le programme sur un éditeur quelconque puis on compile. Compatible Microsoft.

K-SEKA (Assembleur 68000) — **Kuma**
Assembleur, débugger, déssassembleur. Manuel d'origine en français.

MACRO ASSEMBLEUR — **Metacomco**
Par une société anglaise réputée pour la qualité de ses produits.

GST ASSEMBLEUR — **Mcom**
Bel éditeur sous GEM, complète françisation du produit.

DEVPAC ST — **Hisoft**
Assembleur et éditeur sont chargés ensemble en RAM permettant un assemblage rapide. Debugger inclus.

HIPPO C — **Haba**
Premier C disponible sur ST, il se rapproche des C sous Unix. Notice très succinte, possibilité de détruire l'original en faisant la copie de sécurité.

GST C — **Mcom**
Editeur sous GEM, entièrement francisé, conseillé aux débutants.

LATTICE C — **Metacomco**
Un standard réputé dans le monde du C. Un outil de développement attendu.

MEGAMAX C — Megamax
Système de developpement complet pour le ST. Avec documentation détaillée C, Assembleur, Gem. Il comprend également un programme de création d'icones, de menus, de boites de dialogue.

MICRO C SHELL — Beckemeyer
Rapproche le ST du monde Unix.

PASCAL — OSS
Un pascal standard, très complet, accès à GEM, compilation en une passe. Un manuel de près de 300 pages rend le produit très attractif.
Par un éditeur connu pour l'excellence de ses produits sur Atari.

PASCAL — Metacomco
Pascal standard, compilation en une seule passe.

Pascal UCSD — Pecan
Très bon rapport qualité/prix pour un système de développement habituellement onéreux.

Modula 2 — TD1/Micro-Application
Le langage qui monte. Une sorte de Super Pascal. Très bon outil, apprécié par un grand nombre de développeurs sur ST.

4 × FORTH Level 1 — Dragon Group
Premier Forth disponible sur ST, il n'a pas d'accès à GEM.

4 × FORTH Level 2 — Dragon Group
Cette version permet l'accès à GEM ;

FORTH ACCELERATOR — Dragon Group
Optimise la compilation pour les 4 × Forth.

H à D FORTH — Mirage
Basé sur la dernière version de FORTH (Forth 83), il permet l'accès à GEM, au graphisme, aux interfaces MIDI et imprimantes. Calcul en virgule flottante. Rapport qualité/prix imbattable.

COBOL — Philon
ANSI 74-standard COBOL

FORTRAN — Philon

PRO FORTRAN 77 — Prospero
ANSI 78-standard FORTRAN

METALISP — Metacomco
Implémentation complète du LISP de Cambridge. Pour tous ceux intéressés par l'intelligence artificielle.

PROLOG — OSS
Autre langage qui monte dans les applications d'intelligence artificielle.

Les systèmes experts

L'EXPERT — Mindsoft
L'expert est un système expert d'ordre zéro. Il possède trois modes de fonctionnement : Déduction, Induction et expertise (chaînage avant, arrière et mixte).

Les utilitaires

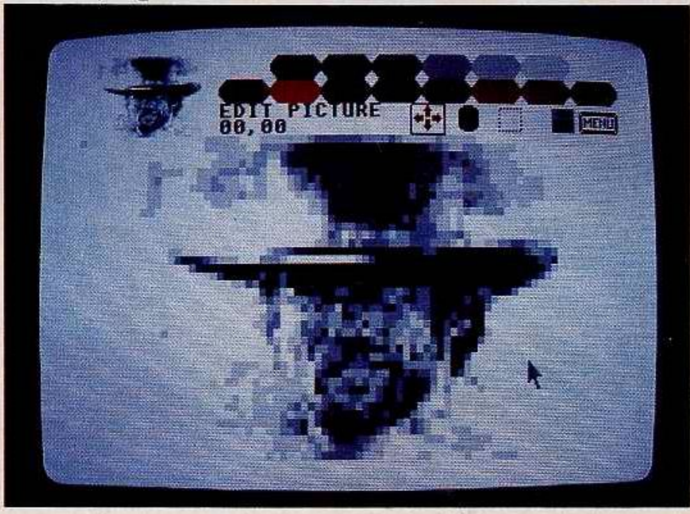
DISQUE VIRTUEL — Pressimage
Ce logiciel permet de créer un pseudo-disque dans la mémoire RAM. Très utile pour tous les programmes faisant de fréquents accès disque.

ST UTILITIES — Hippopotamus
Collection d'utilitaires disques

TOOLBOX — Mirage
Editeur fichier ou secteur. Editeur mémoire RAM. Formattage et copie rapide. Récupération de fichiers perdus. Impression des index du disque.

PERSONNAL DISKIT — OSS
Une collection d'utilitaires pour programmeurs.

C.O.L.R. OBJECT EDITOR — Antic
Créateur d'îcones, éditeur graphique, il crée le code correspondant au graphisme et permet de l'incorporer à un de vos programmes.



C.O.L.R. OBJECT EDITOR

HIPPO PIXEL — Hippopotamus
Générateur de caractères ou générateur de "sprites".

GESTION

Traitement de texte

CODIMP — Pressimage
Ce programme permet d'obtenir la totalité des accents et caractères spéciaux sur écran (circonflexe et tréma) et sur imprimante. Fonctionne avec tout programme étranger si les caractères ASCII supérieurs à 128 n'ont pas été supprimés.
Disponible pour EPSON LX, RX, FX
SMITH CORONA D200 / D300
STAR Gemini 10 / SG 10
OKI 192
CANON PW 1080

ST TEXTE — Atari
Bien que n'étant pas sous GEM, ST TEXTE offre un assez grand nombre de fonctions. Peut relire des textes écrits sous ATARITEXTE sur XL/XE.

LETTER EXPRESS PROCESSOR — Mirage
Plus un traitement de secrétariat qu'un traitement de texte. Comprend un fichier d'adresses.

1st WORD — GST
Traitement de texte sous GEM, il permet de visualiser plusieurs fichiers en même temps sur l'écran, de sélectionner différents types de caractères et d'une manière générale de voir sur écran exactement ce qu'on aura sur imprimante.

HABAWRITER — **Haba**
Traitement de texte entièrement francisé, manuel, messages à l'écran et accentuation. Sous GEM, il permet de travailler sur plusieurs textes à la fois, de visualiser plusieurs polices de caractères et de sauver en ASCII pour utilisation des fichiers avec un autre programme.

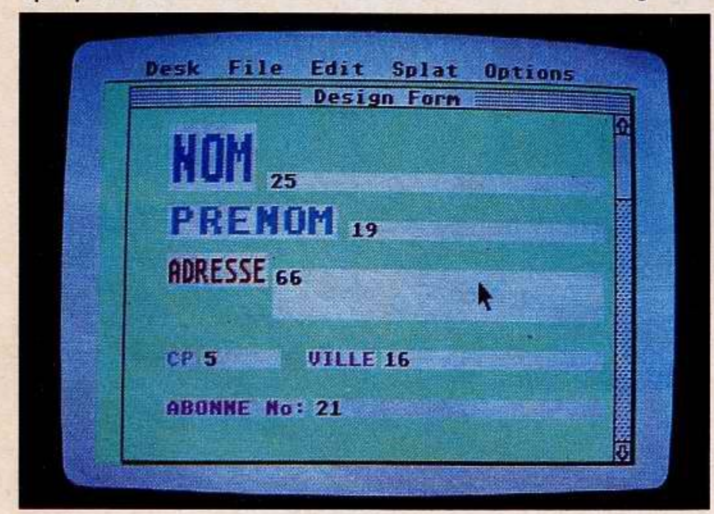
MAILMERGE — **Haba**
Permet de mélanger des données de Habadex et Habawriter.

EVOLUTION — **Loriciels**
Encore peu d'informations sur le premier traitement de texte écrit spécialement pour une utilisation en français. Il s'annonce d'emblée, néanmoins, comme un des plus puissants.

Gestion de fichiers

CARNET D'ADRESSE — **Vrj Soft**
Permet de gerer quelques centaines d'adresses. Possibilité de recherche. Très facile à utiliser.

DB MASTER ONE — **Atari**
Limité à 320K, il charge tout le fichier en mémoire centrale, ce qui le réserve pour des fichiers de moyenne importance. Sa facilité d'utilisation et sa richesse graphique pour l'élaboration des fiches sont un modèle du genre.



DB MASTER ONE

HIPPO SIMPLE — **Hippopotamus**
Peu convivial, mais puissant il permet de trier sur 16 niveaux, de sélectionner sur 16 critères, de faire des calculs intra et inter-fiches. Il sait lire toute RAM additionelle et est entièrement sous GEM.

ZOOMRACKS — **Quickview**
Gestionnaire de fichiers fonctionnant aussi sur IBM, il est supposé être plus logique que les autres dans son fonctionnement.
Il accepte jusqu'à 250 lignes de texte par fiches. Pas sous GEM.

HABADEX — **Haba**
Petit fichier adresse et agenda téléphonique avec possibilité d'appel de numéro automatique. Les adresses peuvent être envoyées vers HABAWRITER grâce à MAILMERGE.

LASERBASE — **Lasersoft**
Connu sur le MacIntosh, Laserbase est très complet. Puissant, riche graphiquement, il n'est pas protégé ce qui rend son utilisation facile sur disque dur.

H & D BASE — **Mirage**
Logiciel largement inspiré de DBASE II, H & D BASE est la première base de données relationnelles sur ST. Attention ! L'emploi de ce genre de logiciel très puissant nécessite un apprentissage.

DBMAN — **Versasoft**
Compatible Dbase II, Dbase III ce logiciel peut relire des applications écrites dans ces langages, sous réserves de légères modifications.

THE MANAGER — **Atari**
Permet d'écrire ses propres applications. Prend toute sa puissance avec le disque dur.

Tableurs

K-SPREAD — **Kuma**
Sous GEM, c'est un tableur utilisable rapidement par un néophite. Puissant par son éditeur, le nombre de lignes (8191) et de colonnes (256). Par contre il n'a ni macros, ni graphisme et seulement les fonctions mathématiques et les formules les plus simples.

VIP PROFESSIONNEL — **Vip Technologies**
Complètement compatible Lotus 1.2.3, VIP est un des logiciels les plus puissants disponible à ce jour. Graphisme, macros, fichiers, tout est compatible 1.2.3 sur un tableau de 8191x256.
Attention! VIP nécessite actuellement un TOS spécial pour fonctionner correctement.

RYTHM — **Softechnies**
Mini tableur en accessoire de bureau.

Graphisme

K-GRAPH — **Kuma**
Création de graphique type camembert, histogramme, ... pour les logiciels K-SPREAD et K-DATA.

Gestion

TIMELINK — **Softechnies**
Agenda sur une année.

PI COMPTA — **Pressimage**
Comptabilite pour petite société facile d'utilisation.

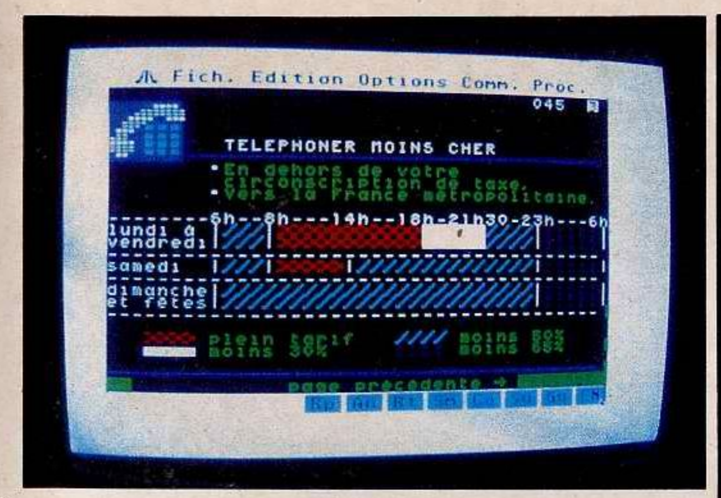
ST FACT — **Pressimage**
Gestion de stock avec comptes clients et facturation. Sous GEM, elle est utilisable par un néophite immédiatement.

LA COMMUNICATION

Teletel

EMULCOM — **Atari**
Logiciel d'émulation Minitel, Emulcom recrée un écran Minitel sur ST en monochrome ou couleur. Les fonctions sont innombrables, la moins intéressante n'étant pas celle qui vous permet de savoir à tout moment ce que vous coûtent vos communications.

LORITEL — **Loriciels**
Disponible sur de nombreux micros, Loritel permet le téléchargement de toutes sortes de fichiers et même de programmes.



EMULCOM

Autres types de communications

K-COM — **Kuma**
Télétype, VT52 et VT100 sont émulés par K-COM. Protocole XModem inclus.

PC/INTERCOM — **Mark of the Unicorn**
Emulateur VT 100 le plus complet.

ST-2392 — **Coincidence**
Emulateur complet pour des systèmes multi-utilisateurs Hewlett-
Packard. Permet toutes les fonctions de l'HP 2392.

LA CREATIVITE

Graphisme

NEOCHROME — **Atari**
Orienté plus sur la couleur que sur le dessin, NEOCHROME ne fonctionne qu'en basse résolution. La palette entière de couleur est visible à chaque instant sur l'écran.

DEGAS — **Batteries Included**
Logiciel de création picturale, DEGAS est l'outil actuel des graphistes sur ST. Il travaille dans les trois résolutions. Un générateur de nouvelles polices de caractères est également inclus.

COLORSPACE — **LLamasoft**
Permet de créer des animations colorées sur l'écran du ST. Compatible avec NEOCHROME.

N-VISION — **Audio-light**
Un grand nombre de menus déroulants lui confère la facilité d'emploi de NEOCHROME et la richesse de fonctions de DEGAS. Egalement disponible sur IBM.

	520 ST	PC AT	MAC	AMIGA
Price	$799	$4675	$1995	$1795
CPU	68000	80286	68000	68000
Speed MHz	8.0	6.0	7.83	7.16
RAM	512k	256k	512k	256k
Keys	95	95	59	89
Mouse	Yes	No	Yes	Yes
Res. Color	640x200	640x200	None	640x200
Res. Mono	640x400	720x350	512x342	640x200
Colors	512	Extra	None	4096
Disk Drive	3.5"	5.25"	3.5"	3.5"
DMA Port	Yes	Yes	No	No
MIDI Port	Yes	No	No	No
Voices	3	1	4	4

*NON-INTERLACED

EASY DRAW — **Migraph**
Permet de dessiner des plans d'habitations, de jardins, d'étudier de nouvelles trames de textile et toute sortes de logos. Puissant outil pour ceux qui ont des besoins de dessins professionnels.

Graphisme animé

THE ANIMATOR — **Microdeal**
Création de dessin animé en couleur sur ST.

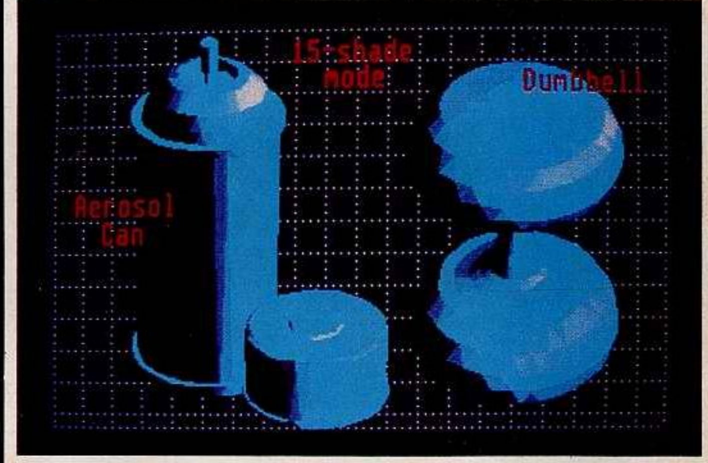
ANIMATE — **Andromeda**
Création de dessin animé encore plus poussé.



ST ART

ST ART — **Andromeda**
Processeur d'images

CAD-3D — **Antic**
Créez et animez vos propres objets en 3D. Squelette ou formes pleines, ombres sur trois sources lumineuses. Zoom.



CAD-3D

Graphisme imprimante

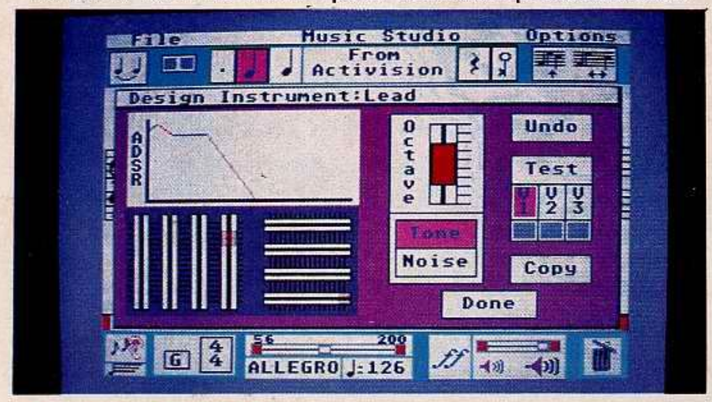
TYPESETTER — **Xlent**
Permet le mélange de graphisme et texte haute résolution sur un grand nombre d'imprimantes.

PRINTMASTER — **Unisound**
Réplique fidèle d'un des plus gros succès logiciel de ces derniers mois. PRINTMASTER reprend toutes les fonctions de PRINT SHOP, mais il permet également de voir à l'écran ce qu'on obtiendra sur le papier.

Musique et son

MUSIC STUDIO — **Activision/Loriciels**
Permet de créer ses propres instruments de musique très facilement en dessinant leur enveloppe sur l'écran. Egalement de maitriser un grand nombre d'instruments de musique par l'interface MIDI.

TWENTY FOUR — **Steinberg**
Veritable outil de travail pour musicien professionnel.



LES JEUX

Arcade/Adresse

MUDPIES — **Michtron**
Petit jeu d'Arcade sorti très tot pour le ST, sa date de sortie (Septembre 85) est un tour de force, mais il est loin d'approcher l'interêt de son grand frère Time Bandit.

MEGAROIDS — **Atari**
Dans le domaine public, ce jeu montre la puissance du langage C pour écrire des jeux d'arcade. Tout simplement la meilleure version jamais réalisé du célèbre »Asteroids»

JOUST — **Atari**
Basé sur le jeu d'arcade du même nom, ce programme a été en grande partie réalisé grâce au logiciel « C.O.L.R Object Editor ».
Peut se jouer à deux, uniquement en couleur.

BATTLEZONE — **Andromeda**
Bataille de char en trois dimensions.

MILLIPEDE — **Andromeda**
La célèbre chenille des jeux d'arcade.

CRISTAL CASTLE — **Andromeda**
Encore un best seller des jeux vidéo.

STAR RAIDERS — **Atari**
La version 68000 de la plus fantastique guerre de l'espace disponible sur micro 8 bits.

TIME BANDITS — **Michtron**
Animation en 3D pour un jeu rivalisant avec ce qui se fait de mieux en Arcade. Graphisme étonnant, créatures et écrans multiples font de ce jeu original un must pour les amateurs de jeux rapides.
Couleur uniquement.

RED ALERT — **Antic**
Jeu type »Wargames» ou vous devez détruire les missiles venus de l'Est. Il existe une option Etats-Unis ou une option Europe.

STARGLIDER — **Rainbird**
Guerre et animation en 3D.

MAJOR MOTION — **Michtron**
Poursuite en voiture.

KUNG FU:THE WAY OF THE EXPLODING FIST
Le jeu de karaté le plus célèbre du moment.

SHAOLIN WARRIOR — **Michtron**
Autre jeu de karaté.

GOLD RUNNER — **Michtron**
Lode runner en 50 tableaux.

Simulation

SILENT SERVICE — **Microprose**
Simulation complète de guerre sous-marine, avec cartes maritimes, vue periscopique, identification des batiments

FLIGHT SIMULATOR II — **Sublogic**
La célèbre simulation aérienne, avec commande d'un petit monoplace, survol possible de tous les Etats-Unis avec le graphisme du ST.

F15 STRIKE EAGLE — **Microprose**
Guerre aérienne au Moyen Orient, vous êtes aux commandes d'un chasseur de combat.

Aventure texte

Tous les jeux d'INFOCOM sont disponibles pour le ST. Très bien placés dans tous les hit-parades aux Etats-Unis, ils nécessitent une grande connaissance de la langue anglaise pour pouvoir être appréciés.

SKUL/WEST — **Talent**
Deux jeux complets en anglais avec un vocabulaire étendu.

Aventure graphique

NINE PRINCES IN AMBER — **Spinnaker (Telarium)**
D'après Roger Zelazny, une aventure à la fois médiévale et futuriste.

AMAZON — **Spinnaker (Telarium)**
Agent d'une société spécialisée en technologie avançée, vous êtes envoyé dans une zone inexplorée de l'Amazone. D'après un roman de Michael Crichton.

DRAGONWORLD — **Spinnaker (Telarium)**
Scientifique et un peu visionnaire, vous êtes le seul à pouvoir sauver le dernier dragon de Fedora.

TREASURE ISLAND — **Spinnaker (Windham)**
Retrouver l'or en échappant à Long John Silver, c'est tout le parfum nostalgique de l'ile au trésor.

THE WIZARD OF OZ — **Spinnaker (Windham)**
Tous les étranges personnages du magicien d'Oz sur la route de la ville d'Emeraude.

TRANSYLVANIA — **Penguin**
Vous voyagez aux confins des Carpathes. Une princesse et des vampires sont au programme.

CRIMSON CROWN — **Penguin**
Dans le même monde étrange que celui de Transylvania, partez à la recherche d'une couronne magique.

BORROWED TIME — **Activision/Loriciels**
Vous avez moins d'une journée pour résoudre un cas difficile et éviter une tentative de meurtre à votre encontre.

FARENHEIT 451 — Spinnaker (Telarium)
D'après le célèbre roman de Bradbury, un classique du jeu d'aventure et best seller sur Apple. Joignez l'Underground sous un état policier du futur.

PERRY MASON — Spinnaker (Telarium)
»The case of the Mandarin murder» est une enquête célèbre d'un fameux détective tout droit venu d'une série télévisée.

MINDSHADOW — Activision/Loriciels
Vous êtes victime d'une amnésie, saurez-vous retrouver qui vous êtes ?

HACKER — Activision/Logiciels
Vous vous introduisez accidentellement dans l'ordinateur d'une grosse société. Vous voilà entrainé dans une sombre histoire d'espionnage.

THE PAWN — Rainbird
Les plus belles images à ce jour dans un jeu d'aventure sur micro. Comme l'ordinateur comprend en outre une syntaxe étendue, c'est un jeu à considérer de très près.



THE PAWN

Aventure rôle

ULTIMA II — Sierra
Le best seller des jeux de rôle, Ultima vous emmène à travers les âges, sur d'autres planètes et met à rude épreuve votre joueur avec un voyage dans un univers fantastique.

Aventure animée

LANDS OF HAVOC — Microdeal
Issu du QL, ce jeu restera dans l'histoire comme le premier jeu sorti sur ST. Des centaines d'écrans servent de fond à une bagarre permanente avec d'étranges créatures.

KING QUEST — Sierra
Premier programme de jeu d'aventure tridimensionnel animé, King Quest a étonné tous les amateurs de jeu d'aventure dans sa version Apple 128K/écran haute résolution. Maintenant disponible sur ST.

KING QUEST II — Sierra
Plus de 100 écrans terrestres ou aquatiques en trois dimensions sont le théatre de votre rencontre avec de multiples créatures . Presque du dessin animé !

THE BLACK CAULDRON — Sierra
Developpé avec l'équipe de Walt Disney, Ce jeu pousse encore plus loin l'animation et la qualité graphique.

BRATACCAS — Psychnosis
Dans une base isolée sur une planète perdue, vous avez maille à partir avec tous les autres occupants. Totalement animé, ce jeu a été écrit spécialement pour le ST et préfigure les nouvelles sortes de logiciels permis par la puissance et la résolution du ST.

Stratégie

HEX — Mark of the Unicorn
Sur un damier en relief multicolore, vous devez user de reflexion pour contrer les agissements d'entités subversives.

FLIP SIDE — Michtron
Jeu classique d'Othello/Reversi, niveau de jeu moyen, graphisme intéressant en couleur ou N/B.

BACKGAMMON — Hippopotamus
Très beau graphisme (Couleur et N/B) pour un jeu trop peu connu. Des principes d'intelligence artificielle ont été introduit dans le jeu de l'ordinateur.

CHESS — I.C.S.
Basé sur le programme Cyruss, un des plus fort programmes d'échecs paru à ce jour. Graphique splendide en couleur et en 3D.



BACKGAMMON

Stratégie animée

SUNDOG — FTL Games
A la fois aventure, guerre de l'espace, simulation économique, Sundog a subi un lifting étonnant par rapport à sa version Apple.
Un des tous meilleurs jeux actuels, avec des graphismes superbes. Uniquement couleur.



Jeux divers

BIORYTHME — Cobrasoft
Un biorythme intéressant pour les programmeurs car livré avec son source.

CHIFFRES ET LETTRES — Atari
Le jeu d'Antenne 2, comme vous ne l'avez jamais vu. Graphisme et puissance. Les champions qui l'ont testés n'ont pu le mettre en défaut.

STAR STRUCK — Antic
Génère des cartes et des thèmes astraux comme un vrai astrologue professionnel.

LES PERIPHERIQUES



VIDEO DIGITIZER — **Hippopotamus**
Ce produit permet la digitalisation de n'importe quel signal vidéo (caméra, magnétoscope, TV) en PAL/SECAM ou NTSC. Les images sont compatibles DEGAS et NEOCHROME. Noir et blanc en haute résolution et 16 nuances de gris en 320 sur 200.

VIDEO DIGITIZER — **Print Technik**
Mêmes caractéristiques, sauve sous format NEOCHROME ou DOODLE, permet impression couleur sur CANON PJ-1080.

SOUND DIGITIZER — **Hippopotamus**
Module de digitalisation sonore. La version 1 méga permet 40 secondes d'enregistrement. Demonstration de reconnaissance vocale.

EPROM BURNER — **Hippopotamus**
Lit, vérifie et écrit la plupart des Eproms du marché en particulier les 2764,27128,27256 et 27512. Copie d'Eproms ou création à partir de fichiers disque.

DISQUE DUR 10 Mega
DISQUE DUR 20 Mega
DISQUE DUR 30 Mega
DISQUE DUR 60 Mega — **Supra**

LES ACCESSOIRES

Rangement

BOITIER DISQUETTE 3'5
De type vidéo, ce boitier permet de présenter et stocker 10 disquettes.

CLASSEUR TOILE 3'5
Ce classeur en toile permet de présenter et stocker 28 disquettes.

BOITE DISQUETTES 3'5
Cette boite en plastique dur permet de présenter et stocker 50 disquettes.

STAND
Ce stand permet de donner un aspect plus professionnel à votre ST. L'arrière de l'unité centrale se glisse dessous. La ou les unités de disquettes se rangent au premier niveau et le moniteur se pose au dessus. Les fils et alimentations sont cachés.

Connectique

CORDON VIDEO PERITEL
Pour relier un ST à une télévision ou un moniteur par la prise Peritel.

CORDON IMPRIMANTE CENTRONICS
Pour relier un ST à une imprimante possédant une interface parallèle Centronics.

CORDON MODEM RS 232
Pour relier un ST à un modem ou tout autre périphérique muni d'une interface série RS 232 avec connecteur standard.

CORDONS MIDI
Pour relier un ST à tout appareil musical possédant une interface MIDI.

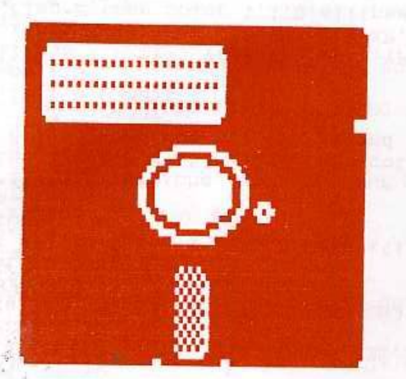
OTHELLO REVERSI EN BASIC

```
1      '*************************************************************
2      '**                                                         **
3      '**   jeu d'OTHELLO en graphisme 3D ecrit en ST BASIC // 1986  **
4      '**        Copyright Chr.Bonnet et ST Magazine              **
5      '**                                                         **
6      '*************************************************************
7      '
10     '**************** initialisation generale et debut ****************
11     dim c(9,9),b(9,9),x(8,8),y(8,8),r(8,8)       :'
15     gosub 20000                                    :'
998    '
1000   '************** initialisation secondaire et partie **************
1020   restore 9990                                   :'
1030   for i=0 to 9:for j=0 to 9                      :'
1035   read a                                         :'
1040   b(i,j)=a                                       :'  donne le code de depart
1045   if i+j<>0 then c(i,j)=1                        :'      de chaque case
1050   next j,i                                       :'   (contenu et valeur)
1055   c(4,4)=2:c(5,5)=2:c(4,5)=3:c(5,4)=3            :'
1060   score(2)=2:score(3)=2                          :'
1070   gosub 2000                                     :'   dessine les 4 pions
1080   gotoxy 24,6:print "Voulez-vous commencer ?"    :'       du debut
1090   gotoxy 24,7:line input a$                      :'          et
1095   gotoxy 24,6:print "                       "    :'      demande qui
1096   gotoxy 24,7:print "                       "    :'   commence la partie
1100   if left$(a$,1)<>"O" or left$(a$,1)<>"o" then goto 1130
1110   phase=2:adv=3                                  :'
1115   passe=0                                        :'     tour du joueur
1120   gosub 3000                                     :'
1125   if score(2)+score(3)=64 then goto 7000         :'
1127   gotoxy 29,10:print score(2);"PIONS     "       :'
1128   gotoxy 29,11:print score(3);"PIONS     "       :'   affichage du score
1130   phase=3:adv=2                                  :'
1140   gosub 4000                                     :'     tour du joueur
1160   if passe=2 or score(phase)+score(adv)=64 then goto 7000
1170   gotoxy 29,10:print score(2);"PIONS     "       :'
1180   gotoxy 29,11:print score(3);"PIONS     "       :'   affichage du score
1997   goto 1110                                      :'
1998   '
1999   '************** place les 4 pions du debut *************************
2000   xp=4:yp=4:flag=2:gosub 30000                   :'     pion 1 (joueur)
2010   xp=5:yp=5:gosub 30000                          :'     pion 2 (joueur)
2020   xp=4:yp=5:flag=3:gosub 30000                   :'     pion 3 (atari )
2030   xp=5:yp=4:gosub 30000                          :'     pion 4 (atari )
2190   gotoxy 24,10:print "VOUS:":gotoxy 24,11:print "MOI :"
2997   return                                         :'
2998   '                                              :'
2999   '************** tour de jeu du joueur humain ***********************
3000   '                                              :'
3010   gotoxy 24,6:print "VOTRE COUP:"                :'
3020   gosub 50000                                    :'
3025   if key=2 then for i=6 to 9:gotoxy 24,I:print "             ":next:passe=1:
return
3040   if v<1 or w<1 or v>8 or w>8 then goto 3020     :' est-il dans le damier?
3050   if c(v,w)<>1 then goto 3020                    :' est-ce une case vide ?
3060   gosub 6000                                     :'  peut-on reourner des
3070   if flip<1 then goto 3020                       :'     pions adverses ?
3080   for i=6 to 8                                   :'
3090   gotoxy 24,i:print "             "              :'     (13 espaces)
3100   next                                           :'
3110   vj=v:wj=w:gosub 5000                           :'   retourne les pions
3997   return                                         :'
3998   rem                                            :'
3999   '**************** tour de jeu du programme ST **********************
4000   rem                                            :'
4010   xcase=0:ycase=0:point=-99                      :' initialise le tour atari
4020   for v=1 to 8                                   :'
4030   for w=1 to 8                                   :'
4040   if c(v,w)<>1 then 4070                         :'   case est-elle vide ?
4050   gosub 6000                                     :'   peut-on retourner au
4051   if flip=0 then 4070                            :' moins 1 pion adverse ?
4052   if score(2)+score(3)<34 then flip=-flip        :'
4053   if score(2)+score(3)>56 then flip=flip*6       :' affecte une valeur en
4055   valeur=(b(v,w)*3)+flip                         :'   fonction du moment
4057   if b(v,w)=-4  then gosub 10000                 :'
4058   if b(v,w)=-12 then gosub 11000                 :'
4060   if valeur>point then point=valeur:xcase=v:ycase=w
4062   IF valeur=point and rnd(3)<0.6 then point=valeur:xcase=v:ycase=w
4065   gotoxy 24,13:print "MEILLEURE CASE:";xcase;ycase
4066   gotoxy 24,14:print "VALEUR        :";point    :'
4067   gotoxy 24,16:print "CASE ETUDIEE  :";v;w      :'
4068   gotoxy 24,17:print "VALEUR        :";valeur   :'
4070   next w,v                                       :'
```

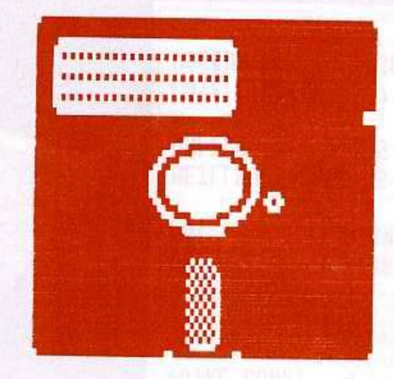
23 espaces

```
4080  if point=-99 then passe=passe+1:goto 4997   :'  pas de coup jouable
4090  vj=xcase:wj=ycase:gosub 5000                :'  joue le coup choisi
4997  return                                      :'  fin du tour atari
4998  rem
4999  '************ retourne les pions sur l'ecran et dans les variables ***
5000  rem                                         :'
5010  flag=phase:xp=vj:yp=wj:gosub 30000:c(xp,yp)=phase
5025  score(phase)=score(phase)+1                 :'  un pion de plus.
5030  gosub 8000                                  :'  modifie la valeur des
5040  for i=-1 to 1                               :'        cases
5050  for j=-1 to 1                               :'
5060  if i=0 and j=0 then goto 5170               :'  regarde autour de la
5070  flop=0:dv=vj:dw=wj                          :'      case jouee
5080  dv=dv+i:dw=dw+j                             :'
5090  if c(dv,dw)<2 then goto 5170                :'  retourne les pions
5100  if c(dv,dw)=adv then flop=flop+1:goto 5080  :'  adverses possibles
5110  if flop<1 then goto 5170                    :'  autour de la case jouee
5120  dv=dv-i:dw=dw-j                             :'
5130  if dv=vj and dw=wj then goto 5170           :'
5140  c(dv,dw)=phase                              :'
5145  flag=phase:xp=dv:yp=dw:gosub 30000          :'  retourne ce pion-la
5150  score(phase)=score(phase)+1:score(adv)=score(adv)-1
5160  goto 5120                                   :'
5170  next j,i                                    :'  passe a la direction
5997  return                                      :'      suivante
5998  rem                                         :'
5999  '****** verifie a l'avance si on peut retourner au moins 1 pion ******
6000  '                                           :'
6010  flip=0:valeur=0                             :'  flip=nbre de pions
6020  for i=-1 to 1                               :'  retournes au total
6030  for j=-1 to 1                               :'
6040  if i=0 and j=0 then goto 6110               :'
6050  flop=0:dx=v:dy=w                            :'  flop=nbre de pions
6060  dx=dx+i:dy=dy+j                             :'  retournes dans une
6070  if c(dx,dy)<2 then goto 6110                :'      direction
6080  if c(dx,dy)=phase then goto 6100            :'
6090  flop=flop+1:goto 6060                       :'  un pion de plus...
6100  if flop>0 then flip=flip+flop               :'
6110  next j,i                                    :'
6997  return                                      :'  direction suivante
6998  rem                                         :'
6999  '************ fin de la partie et attente d'une option ***************
7000  '                                           :'
7010  gotoxy 24,10:print "VOUS:";score(2);"PIONS.      "    :'  score
7020  gotoxy 24,11:print "MOI :";score(3);"PIONS.      "    :'  final
7030  gotoxy 24,8                                 :'
7040  if score(2)=score(3) then print "EGALITE !!":goto 7100
7050  if score(2)>score(3) then print "VOUS AVEZ GAGNE...BRAVO":goto 7100
7060  print "J'AI GAGNE!"                         :'
7100  gosub 40100:goto 15                         :'
7998  rem                                         :'
7999  '********** modifie les valeurs des cases en fonction du pion joue ****
8000  rem                                         :'
8010  if b(vj,wj)<>16 then goto 8100              :'  est-ce un coin ?
8020  for i=vj-2 to vj+2:if i<0 or i>8 then goto 8045
8025  for j=wj-2 to wj+2:if j<0 or j>8 then goto 8040
8030  if b(i,j) < -3 then b(i,j)=0                :'  modifie la valeur des
8040  next j                                      :'  cases autour d'un coin
8045  next i                                      :'  si celui-ci est deja
8050  return                                      :'      occupe
8100  if b(vj,wj)<>5 then goto 8200               :'  est-ce un bord ?
8110  for i=vj-1 to vj+1:for j=wj-1 to wj+1       :'
8120  if b(i,j)=1 then b(i,j)=3                   :'  modifie la valeur des
8130  if b(i,j)=-2 then b(i,j)=1                  :'  cases si la case-bord
8140  next j,i                                    :'  etudie est occupee...
8200  rem                                         :'
8997  return                                      :'
8998  rem                                         :'
8999  '*********** initialise les valeurs des cases *********************
9989  rem                                         :'
9990  DATA 0,0,0,0,0,0,0,0,0,0
9991  DATA 0,16,-4,5,1,1,5,-4,16,0
9992  DATA 0,-4,-12,-2,-2,-2,-2,-12,-4,0
9993  DATA 0,5,-2,4,2,2,4,-2,5,0
9994  DATA 0,1,-2,2,1,1,2,-2,1,0
9995  DATA 0,1,-2,2,1,1,2,-2,1,0
9996  DATA 0,5,-2,4,2,2,4,-2,5,0
9997  DATA 0,-4,-12,-2,-2,-2,-2,-12,-4,0
9998  DATA 0,16,-4,5,1,1,5,-4,16,0
9999  DATA 0,0,0,0,0,0,0,0,0,0
10000 rem
10001 '****** teste si la case donne un coin a l'adversaire *******
10010 dv=0:dw=0:flag=-80                          :'
10020 if v=2 then dv=1                            :'  choisit des directions
10030 if v=7 then dv=-1                           :'      a etudier...
10040 if w=2 then dw=1                            :'
10050 if w=7 then dw=-1                           :'
10060 pv=v:pw=w                                   :'
10070 pv=pv+dv:pw=pw+dw                           :'
10080 if c(pv,pw)=0 or c(pv,pw)=1 then return     :'
10090 if c(pv,pw)=3 then goto 10070               :'
10100 if pv-dv<>v or pw-dw<>w then valeur=flag:return
10110 pv=pv+dv:pw=pw+dw                           :'
10120 if c(pv,pw)=0 or c(pv,pw)=1 then valeur=flag:return
10130 if c(pv,pw)=3 then return                   :'
10140 goto 10110                                  :'
10999 rem
11000 '***** teste si la case (-12) donne un coin a l'adversaire ****
11010 dv=0:dw=0:flag=-60                          :'
11020 if v=2 then dv=1 else dv=-1                 :'  choix de la direction
11030 if w=2 then dw=1 else dw=-1                 :'      a etudier
11040 goto 10060                                  :'
19998 rem                                         :'
19999 '***************** Dessin de plateau relief **** BC 1985 **********
20000 '** INITIALISE L'ECRAN ET LES VARIABLES DE DESSIN *****************
20010 fullw 2:clearw 2:color 1,0,0,1,1:gotoxy 10,9:randomize (peek(1213)):clear
20015 xx=0:x1=639:yy=19:y1=36:gosub 40050:yy=385:y1=399:gosub 40050:xx=618:x1=63
9:yy=19:y1=399:gosub 40050
20017 xx=60:x1=639:yy=0:y1=17:gosub 40050         :'  remarque:
20020 color 1,1,1,1,1:linef 10,100,240,69         :'  le PEEK(1213) a la ligne
20030 LINEF 240,69,360,230                        :'  20010 prend la valeur
20040 LINEF 360,230,100,300                       :'  de l'horloge interne
20050 LINEF 100,300,10,100                        :'  pour assurer un "vrai"
20060 COLOR 1,1,1,4,2                             :'  RANDOM...
20070 FILL 100,200
20080 LINEF 216,72,330,238                        :'  dessine le contour du
20090 LINEF 188,74,302,246                        :'      plateau de jeu
20100 LINEF 164,78,268,254                        :'
20110 LINEF 136,82,238,264                        :'
20120 LINEF 106,86,206,272                        :'
20130 LINEF 74,92,172,280                         :'
20140 LINEF 42,96,136,290                         :'
20150 REM ----------- LIGNES HORIZONTALES ------------------------------------
20160 LINEF 20,120,254,86                         :'
20170 LINEF 30,142,266,104                        :'
20180 LINEF 40,164,280,124                        :'
20190 LINEF 50,188,296,144                        :'
20200 LINEF 62,214,310,164                        :'
20210 LINEF 74,242,326,184                        :'
20220 LINEF 86,270,342,206                        :'
```

```
20230 REM ----------- EPAISSEUR -----------------------------------------------
20240 LINEF 360,230,360,240                 :'
20250 LINEF 360,240,100,312                 :'
20260 LINEF 100,312,100,300                 :'
20270 LINEF 100,312,10,110                  :'
20280 LINEF 10,110,10,100                   :'
20290 COLOR 1,1,1,2,2                       :'
20300 FILL 110,305                          :'
20310 COLOR 1,1,1,3,2                       :'
20320 FILL 98,300                           :'
20330 REM -----------------CADRE -----------------------------------------------
20340 LINEF 400,10,600,10                   :'
20350 LINEF 600,10,600,330                  :'
20360 LINEF 600,330,400,330                 :'
20370 LINEF 400,330,400,10                  :'
20380 LINEF 395,5,605,5                     :'
20390 LINEF 605,5,605,335                   :'
20400 LINEF 605,335,395,335                 :'
20410 LINEF 395,335,395,5                   :'
20420 '***************** TEXTE ***********************************************
20425 POKE CONTRL,106:POKE INTIN,7:VDISYS(1)  :'
20430 TAILLE=99:GOSUB 40000:GOTOXY 26,2:PRINT "O":GOTOXY 27,2:PRINT "T":GOTOXY 2
8,2:PRINT "H":GOTOXY 29,2:PRINT "E":GOTOXY 30,2:PRINT "L":GOTOXY 31,2:PRINT "L":
GOTOXY 32,2:PRINT "O"
20435 POKE CONTRL,106:POKE INTIN,0:VDISYS(1)  :'
20440 GOTOXY 24,3:TAILLE=04:GOSUB 40000:PRINT "Par Chr.BONNET    ";CHR$(189);" 19
86"
20450 COLOR 1,1,1,8,2:TAILLE=13:GOSUB 40000   :'
20460 FILL 0,0                                :'
20470 REM ----------- PIONS -----------------------------------------
20474 rem
20475 rem        3 valeurs sont affectees a chaque case:la position du centre
20476 rem        en X et Y, et le rayon du pion pour cette case. ex: X,Y,R
20477 rem
20480 data 32,108,14,63,104,13,96,100,13,126,94,13,156,90,12,182,86,12,208,83,11
,234,80,11
20490 data 42,130,14,74,124,14,106,120,14,138,115,13,166,110,12,194,105,12,222,1
00,11,247,97,11
20500 data 52,150,14,84,145,14,118,140,14,148,135,14,178,130,13,206,125,13,234,1
20,12,260,115,11
20510 data 61,172,15,97,168,14,128,162,14,160,156,14,191,150,13,219,144,13,246,1
40,12,274,136,12
20520 data 74,198,15,109,192,15,142,185,15,174,180,14,204,172,14,233,166,13,262,
160,13,290,156,12
20530 data 86,224,16,122,216,16,155,210,15,186,202,14,217,196,14,247,188,14,276,
182,13,305,176,12
20540 data 98,252,17,135,244,16,168,235,16,200,228,15,232,220,14,262,213,14,292,
205,13,320,198,13
20550 data 112,280,18,148,271,17,182,263,16,214,254,15,246,245,15,278,238,14,307
,230,14,336,222,13
20560 restore 20480                           :'
20570 for j=1 to 8:for i=1 to 8:read x(i,j),y(i,j),r(i,j):next i,j
20575 return                                  :'
29998 rem                                     :'
29999 '********* affiche les pions sur l'ecran *************************
30000 color 1,1,1,8,2:if flag=3 then color 1,1,1,1,5
30005 if flag=3 then for b=1 TO 7:sound 1,15,8,8,0:next else for b=7 to 1 step -
                                                                               1
:sound 1,15,b,b,0:next
30006 sound 1,0                               :'
30010 pellipse X(XP,YP),Y(XP,YP),R(XP,YP),(R(XP,YP)/2)
30020 pellipse X(XP,YP),Y(XP,YP)-4,R(XP,YP),(R(XP,YP)/2)
30030 if flag=2 then color 1,1,0:ellipse X(XP,YP),Y(XP,YP)-4,R(XP,YP),(R(XP,YP)/
2)
30040 return                                  :'
39999 rem                                     :'
40000 '************** TAILLE DES CARACTERES ** (routine VDI) ***********
40010 poke contrl,12:poke contrl+2,1:poke contrl+4,1:poke contrl+6,1:poke
contrl+ 8,1:poke contrl+10,1
40020 poke ptsin,1:poke ptsin+2,taille:poke ptsin+4,1:poke ptsin+6,1:poke
ptsin+8,1:poke ptsin+10,1
40030 vdisys(1)
40040 return
40049 rem
40050 '************* RECTANGLE AFFICHE ******* (routine VDI) *********
40060 poke contrl,11:poke contrl+2,2:poke contrl+6,0:poke contrl+10,1:poke
ptsin,xx:poke ptsin+2,yy:poke ptsin+4,xl:poke ptsin+6,yl
40070 vdisys(1)
40080 return
40089 rem
40090 '************** COORDONNEES SOURIS ***** (routine AES) *********
40100 A#=GB:CONTROL=PEEK(A#):GLOBAL=PEEK(A#+4):GINTIN=PEEK(A#+8):GINTOUT=
PEEK(A#+12):ADDRIN=PEEK(A#+16):ADDROUT=PEEK(A#+20)
40110 GEMSYS(79)
40120 XM=PEEK(GINTOUT+2):YM=PEEK(GINTOUT+4):KEY=PEEK(GINTOUT+6)
40130 if key=0 then 40100
40140 return
49998 rem
49999 '*********** routine de saisie de la souris ** d'apres Manuel Atari **
50000 gosub 40100:xm=xm-1:ym=ym-38         :'
50005 v=0:w=0                              :'
50010 for i=1 to 8:for j=1 to 8            :' regarde si le curseur de
50020 dd=sqr(((x(i,j)-xm)^2)+((y(i,j)-ym)^2))  :' la souris se trouve bien
50030 if dd<r(i,j) then v=i:w=j            :'    dans une des cases du
50040 next j,i                             :' plateau de jeu et renvoie
50050 return                               :' la position de cette case
```



PROGRAMMATION DU MFP68901



Dans ce deuxième article sur le circuit multifonction de Motorola, nous allons examiner de façon pratique comment s'effectue la programmation des Timers et l'utilisation du gestionnaire d'interruption.

Pour cela les concepteurs du système d'exploitation ont prévu des fonctions qui nous facilitent la tâche pour ce qui est de l'utilisation de ce circuit. A cet effet les routines systèmes qui interfacent le système d'exploitaion et le hardware de la machine (le ST BIOS) contiennent des fonctions qui permettent la programmation des Timers et la gestion des interruptions. Pour une bonne compréhension de cette étude nous allons considérer les quelques routines du ST BIOS qui nous intéressent, les expliquer et enfin les utiliser pour la programmation du Timer A. Un programme d'application nous permettra à cet effet de programmer le Timer A en interruption et d'afficher toutes les 5 secondes un messages sur l'écran. Cet exemple d'application fera l'objet du prochain numéro de ST magasine.

Avant de commencer je dois faire une rectification sur l'article du précédent numéro qui proposait de modifier le Timer C pour une utilisation personnelle. En réalité, il s'agit du Timer A, le C étant un Timer utilisé par le système pour générer une fréquence de 200 Hz utilisable par interruption dans les applications qui en trouveraient l'utilité.

Examinons maintenant les routines de l'eXtended BIOS qui nous permettent de programmer l'un quelconque des Timers A, B, C ou D. Rappelons que les fonctions du XBIOS sont appelables par la Trap 14.

La fonction qui programme les Timers du MFP68901 est appelé xbtimer. Les paramètres que l'on doit communiquer par l'intermédiaire de cette fonction sont dans l'ordre :

- id (0 à 3 pour les Timers A à D)

- control (valeur à placer dans le registre de controle du Timer sélectionné)

- data (valeur à placer dans le regsitre de données du Timer sélectionné)

- intvec (est l'adresse de la routine d'interruption vers laquelle est dirigé le programme une fois que le Timer a fait un cycle de compatge ou de décomptage)

Pour ceux qui possèdent le kit de développement l'appel de cette fonction se fait en C sous la forme :

```
xbtimer (id,control,data,intvec)
int id,control,data;
long intvec;
```

En assembleur le passage par la fonction 31 de la Trap 14 permet d'accéder aux routines de programmation des timers. Ainsi l'équivalent de cette fonction xbtimer sera

```
move.l   intvec,-(A7)    * adresse routine d'interruption
move.w   data,-(A7)
move.w   control,-(A7)
move.w   id,-(A7)
move.w   #31,-(A7)       * fonction xbtimer
trap     #14             * appel de XBIOS
addq.l   #12,A7          * mise à jour pointeur de pile
```

en assembleur. Vous constatez que le passage des paramètres en assembleur se fait dans l'ordre inverse de celui en C. En effet l'appel des fonctions du XBIOS (et plus généralement du GEMDOS) se fait par la pile (registre A7, ou SP), et cette pile a une structure LIFO (Last In First Out, qui sigifie dernier entré premier sortie). De ce fait le dernier paramètre qui est placé sur la pile (c'est-à-dire 'id') est bien celui qui sera le premier dépilé par la fonction xbtimer, et le premier (c'est-à-dire 'intvec') sera le dernier utilisé par xbtimer.

Remarquez également le format des différents paramètres : 'intvec' est un mot long (32 bits), et 'data', 'control' et 'id' sont bien des mots (16 bits).

Notez enfin la mise à jour de la pile qui se fait par le programmeur en faisant addq.l #12,A7.

La programmation du Timer A se fait alors de la façon suivante :

```
move.l   intvec,(-A7)
move.w   data,-(A7)
move.w   control,-(A7)
move.w   #0,(-A7)          0 => Timer A
move.w   #31,-(A7)
trap     #14
addq.l   #12,A7
```

Ceux d'entre vous qui ne possèdent pas le kit de développement peuvent bien entendu utiliser les routines décrites en assembleur.

La fonction xbtimer se trouve dans le listing du ST BIOS (ce listing est fourni avec le kit de développement). Elle est décrite comme ci-dessous :

```
xbtimer
        moveq     #0,d0       * mise à zéro des registres d0
        moveq     #0,d1       * d1
        moveq     #0,d2       * et d2
        move.w    $4(SP),d0   * id est chargé dans d0
        move.w    $6(SP),d1   * control est chargé dans d1
        move.w    $8(SP),d2   * data est chargé dans d2
        bsr       setimer     * programmer Timer (ci-dessous)
        tst.l     $A(SP)      * vérifie que le vecteur
*                               d'interruption est situé dans
                              * une zone raisonnable
        bmi.b     xbtexit     * ainsi si le contenu de
*                               intvec est > $7fffffff alors
*                               fin
        movea.l   $A(SP),a2   * sinon charger intvec dans a2
        moveq     #0,d1       * mise à zéro de d1
        lea       xbtim,a1    * pointe sur la table des 4
*                               niveaux de priorité des Timers
        andi.l    #$ff,d0     * masque les 3 octets de poids
*                               fort du registre
        move.b    0(a1,d0),d0 * place le niveau d'inter-
*                               ruption du timer sélection-
*                               né dans d0
        bsr       initint     * programmer interruption MFP
xbtexit rts                   * retour
xbtim                         * table des niveaux d'inter-
                              * ruption des 4 timers
          dc.b    13,8,5,4
```

Arrivé à ce point je dois vous donner quelques précisions sur la dernière ligne de cette routine. En effet vous y voyez apparaître 4 nombres qui définissent les 4 niveaux de priorité des Timers du MFP68901. Ce circuit peut en réalité générer un vecteur différent pour 16 sources d'interruptions différentes.

Avant de passer à la suite du listing des fonctions 'setimer' et 'initint' il serait bien de donner une suite théorique au fonctionnement en mode interruption du MFP68901. Ainsi en lisant la suite et plus particulièrement le listing de 'initint' vous comprendrez le pourquoi des choses.

Dans le dernier article nous avons vu que le MFP possède 24 registres. Les adresses de ces différents registres par rapport à l'adresse de base du circuit MFP sont

```
mfp   equ  $fffffa00 * adresse de base du MFP
fonction port
gpip  equ  mfp+1     * registre général d'E/S des données
aer   equ  mfp+3     * registre sélection du front actif
ddr   equ  mfp+5     * registre de direction des données
fonction interruption
iera  equ  mfp+7     * registre validation des interruptions A
ierb  equ  mfp+9     * registre validation des interruptions B
ipra  equ  mfp+11    * reg. d'interruptions en attente A
iprb  equ  mfp+13    * reg. d'interruptions en attente B
isra  equ  mfp+15    * reg. d'interruptions en cours A
isrb  equ  mfp+17    * reg. d'interruptions en cours B
imra  equ  mfp+19    * registre masque d'interruptions A
imrb  equ  mfp+21    * registre masque d'interruptions B
vr    equ  mfp+23    * registre vecteur
fonction timer
tacr  equ  mfp+25    * registre de controle du timer A
tbcr  equ  mfp+27    * registre de controle du timer B
tcdcr equ  mfp+29    * registre de controle des timers C et D
tadr  equ  mfp+31    * registre de données du timer A
tbdr  equ  mfp+33    * registre de données du timer B
tcdr  equ  mfp+35    * registre de données du timer C
tddr  equ  mfp+37    * registre de données du timer D
fonction USART
scr   equ  mfp+39    * registre caractère de synchro.
ucr   equ  mfp+41    * registre de controle de l'USART
rsr   equ  mfp+43    * registre d'état du récepteur
tsr   equ  mfp+45    * registre d'état du transmetteur
udr   equ  mfp+47    * registre de données de l'USART
```

Parmi ces registres 9 sont affectés à la fonction interruption. Ce sont des registres 8 bits qui sont nommés :

- IERA et IERB (registres de validation A et B)
- IPRA et IPRB (registres d'interruption en attente de service)
- ISRA et ISRB (registres d'interruption en cours de service)
- IMRA et IMRB (registres masque des interruptions)
- VR (registre vecteur)

Les registres IERA et IERB valident les canaux d'interruption en positionnant les bits correspondant à 1.

Lorsqu'un canal d'interruption reçoit un signal de demande d'interruption, le bit correspondant, sera mis à 1 dans l'un des registres IPRA IPRB (sauf si le canal a été inhibé par la mise à 0 du bit correspondant dans l'un des registres IERA IERB).

Pour ce qui est des registres IMRA et IMRB, ils servent à masquer les interruptions sur un ou plusieurs canaux. Cette fonction est différente de celle des registres IERA et IERB, qui valident ou inhibent les interruptions; dans ce cas une interruption survenant sur un canal inhibé ne produit en aucune façon ni la mise à 1 du bit correspondant dans les registres IPRA IPRB ni a fortiori la génération d'un signal IRQ à la sortie de la broche 32 du MFP, signal destiné au 68000 (IRQ va en réalité vers le circuit GLUE qui se charge en suite de transmettre les interruptions au microprocesseur). Parcontre le fait de masquer une interruption signifie qu'en cas de demande d'interruption il n'y aura pas de IRQ à la sortie mais néanmoins le bit correspondant dans les registres IPRA IPRB sera positionné jusqu'au traitement approprié. On peut dire que la demande d'interruption est donc mémorisée.

Le registre vecteur VR permet de vectoriser les interruptions. A cet effet les 16 canaux d'interruption du MFP possédent leur propres vecteurs d'interruption. Lorsque le microprocesseur répond à une demande d'interruption par le signal IACK à l'état bas, le MFP peut répondre par un vecteur sur 8 bits. Pour coder 16 canaux il faut 4 bits: les 4 bits de poids faible de ce registre contiennent le numéro du canal. Ces 4 bits représentent la priorité du canal. Quant aux 4 bits de poids fort, il s'agit des 4 bits de poids fort du registre vecteur VR. Le bit VR[3] du registre vecteur sert à définir le mode de fin de traitement d'interruption. Dans le système ST ce bit est positionné à 1 ce qui signifie qu'en cas d'interruption le bit du registre ISRA/ISRB correspondant au canal demandant l'interruption est positionné à 1 alors que le bit correspondant du registre IPRA/IPRB est positionné à 0 lorsque le MFP distribue le vecteur d'interruption. La remise à zéro du bit correspondant du regsitre ISRA/ISRB devra etre effectuée par le programmeur juste avant l'instruction de retour (RTE).

La différence avec le cas où le bit VR[3] est placé à 0 est dans le fait que dans cette dernière situation c'est le MFP lui-meme qui positionnera le bit correspondant du registre IPRA/IPRB à zéro dès qu'il recevra le signal IACK. Le problème dans ce cas est qu'il n'y a plus aucune trace de l'interruption qui est en cours de traitement. A cet effet, dans le mode où VR[3] est à 1, c'est-à-dire le mode du paragraphe précédent, aucun canal de priorité inférieur ne sera servi (néanmoins le bit correspondant du registre IPRA/IPRB sera positionné à 1 pour un traitement en fin du traitement d'interruption en cours), par contre le traitement des interruptions de priorité supérieure sera entrepris.

Avant de finir avec cette partie théorique nous devons préciser les priorités d'interruption du MFP68901 qui peuvent etre décrites par le tableau ci-dessous :

Canal d'interruption	degré	B3	B2	B1	B0
Ligne d'interruption I7	15	1	1	1	1
Ligne d'interruption I6	14	1	1	1	0
TIMER A	<u>13</u>	1	1	0	1
Tampon Récepteur Plein	12	1	1	0	0
Erreur de Réception	11	1	0	1	1
Tampon Transmetteur Vide	10	1	0	1	0
Erreur de Transmission	9	1	0	0	1
TIMER B	<u>8</u>	1	0	0	0
Ligne d'interruption I5	7	0	1	1	1
Ligne d'interruption I4	6	0	1	1	0
TIMER C	<u>5</u>	0	1	0	1
TIMER D	<u>4</u>	0	1	0	0
Ligne d'interruption I3	3	0	0	1	1
Ligne d'interruption I2	2	0	0	1	0
Ligne d'interruption I1	1	0	0	0	1
Ligne d'interruption I0	0	0	0	0	0

ce sont ces 4 bits qui constituront les 4 LSB des vecteurs d'interruptions.

Vous devriez maintenant comprendre la signification de la dernière ligne du programme en assembleur qui définissait les niveaux d'interruption des timers A, B, C et D. A cet effet vous pouvez comparer les valeurs soulignées de la table ci-dessus avec les valeurs devant l'étiquette 'xbtim'.

Quant au registre VR il sera ainsi :

VR[7]	VR[6]	VR[5]	VR[4]	VR[3]	VR[2]	VR[1]	VR[0]
le MSB du vecteur d'inter. initialisé par l'utilisateur				0 ou 1	X	X	X

Passons maintenant à la suite de notre listing. Accrochez-vous bien; à certains endroits le listing nécessite quelque peu de gymnastique, mais une fois le principe assimilé cela devient assez simple :

```
setimer
        movem.l  d0-d4/a0-a3,-(sp)   * sauvegarde de
*                                      registres
        move.l   #$fffffa01,a0       * adresse de base du MFP
*                                      +1 dans a0
        move.l   #imrt,a3            * a3 pointe maintenant sur
*                                      la table des offsets
*                                      permettant d'accéder
*                                      respectivement aux
*                                      registres IMRA, IMRA,
*                                      IMRB et IMRB (voir note
*                                      1)
        move.l   #imrmt,a2           * a2 pointe maintenant sur
*                                      la table des masques
*                                      d'interruption pour les
*                                      timers A, B, C et D
*                                      (voir note 2)
        bsr.b    mskreg              * routine qui masque les
*                                      registres (voir ci-
*                                      dessous)

        move.l   #iert,a3            * accès à IERA, IERA, IERB
*                                      et IERB
        move.l   #iermt,a2           * accès masque des bits de
*                                      validation d'inter-
*                                      ruption des timers
        bsr.b    mskreg              * routine de masquage

        move.l   #iprt,a3            * accès à IPRA, IPRA, IPRB
*                                      et IPRB
        move.l   #iprmt,a2           * accès masque des bits
*                                      d'interruptions en
*                                      attente des timers
        bsr.b    mskreg              * routine de masquage

        move.l   #isrt,a3            * accès à ISRA, ISRA, ISRB
*                                      et ISRB
        move.l   #isrmt,a2           * accès masque des bits
*                                      d'interruptions en cours
*                                      des timers
        bsr.b    mskreg              * routine de masquage
```

```
        move.l   #tcrtab,a3       * accès à TACR, TBCR,
*                                   TCDCR et TCDCR
        move.l   #tcrmsk,a2       * accès masques de 'Stop'
*                                   des timers (voir note 3)
        bsr.b    mskreg           * routine de masquage

        exg      a3,a1            * sauvegarde temporaire de
*                                   a3 dans a1
        lea      tdrtab,a3        * a3 pointe sur la table
*                                   des adresses des
*                                   registres TADR, TBDR,
*                                   TCDR et TDDR
        moveq    #0,d3            * mise à zéro d3
        move.b   0(a3,d0),d3      * regardez la table
*                                   tdrtab : d3 contient
*                                   alors soit 30 (si d0 = 0
*                                   càd timer A à régler)
*                                   soit 32 (si d0 = 1 càd
*                                   timer B) soit 34 (si d0
*                                   = 2 càd timer C) soit 36
*                                   (si d0 = 3 càd timer D).
*                                   Il s'agit de l'adresse
*                                   du registre de données
*                                   du timer sélectionné par
*                                   rapport à l'adresse de
*                                   base du MFP + 1
verify  move.b   d2,0(a0,d3)      * la donnée de d2 est
*                                   placée dans le registre
*                                   de données du timer
*                                   sélectionné (rappel: au
*                                   moment de l'appel de la
*                                   fonction xbtimer le
*                                   paramètre 'data' est
*                                   placé dans d2
        cmp.b    0(a0,d3),d2      * on vérifie que la
*                                   programmation du
*                                   registre a été
*                                   correctement effectuée
*                                   (cette vérification est
*                                   nécessaire d'après le
*                                   data sheet du MFP) en
*                                   comparant la valeur du
*                                   registre avec celle de
*                                   d2
        bne.b    verify           * si pas identiques alors
*                                   on recommence

        exg      a3,a1            * la valeur sauvegardée de
*                                   a3 lui est restituée
        or.b     d1,(a3)          * le paramètre 'control'
*                                   est placé dans le
*                                   registre de controle du
*                                   timer sélectionné

        movem.l  (sp)+,d0-d4/a0-a3 * restitution des
*                                   regsitres sauvegardés
*                                   par l'appel de
*                                   'xbtimer'
        rts                       * retour de ss-programme

mskreg
        bsr.b    getmask          * détermine le masque
*                                   correspondant au timer
*                                   et au registre
*                                   sélectionnés
        move.b   (a2),d3          * le masque est chargé
*                                   dans d3
        and.b    d3,(a3)          * ce ET LOGIQUE masque le
*                                   registre adéquat du MFP
        rts                       * retour

getmask
        moveq    #0,d3            * mise à zéro
        adda     d0,a3            * suivant le numéro du
*                                   timer sélectionné (0,1,2
*                                   ou 3), a3 pointe sur
*                                   l'élément correspondant
*                                   au timer dans l'une des
*                                   tables étiquetées imrt,
*                                   iert, iprt, isrt ou
*                                   tcrtab (revoir les
*                                   instructions du début de
*                                   'setimer')
        move.b   (a3),d3          * chargement de la valeur
*                                   de l'offset pointé par
*                                   a3 dans d3
        add.l    a0,d3            * a0 contenant l'adresse
*                                   de base + 1 du MFP, le
*                                   fait de lui ajouter
*                                   l'offset permet de
*                                   pointer sur l'adresse
*                                   absolue du registre à
*                                   masquer
        movea.l  d3,a3            * cette valeur absolue est
*                                   mise dans a3 afin que
*                                   l'on puisse accéder à
*                                   son contenu en utilisant
*                                   (a3) dans la troisième
*                                   ligne de 'mskreg'
        adda     d0,a2            * de la meme manière a2
*                                   pointe sur la valeur du
*                                   masque, qui sera
*                                   utilisée à la deuxième
*                                   ligne de 'mskreg'
        rts                       * retour

iert    dc.b     6,6,8,8          * offset des adresses des
*                                   registres IERA et IERB
*                                   par rapport à l'adresse
*                                   de base + 1 du MFP (on a
*                                   deux fois 6 et deux fois
*                                   8 puisque IERA (offset
*                                   6) est utilisé pour le
*                                   timer A et le timer B
*                                   et IERB (offset 8) est
*                                   utilisé pour le timer C
*                                   et le timer D). Voir
*                                   également la note 1
iprt    dc.b     10,10,12,12      * offset des adresses des
*                                   registres IPRA et IPRB
*                                   (pour détails, voir
*                                   'iert')
```

```
isrt    dc.b    14,14,16,16     * offset des adresses des
*                                 registres ISRA et ISRB
*                                 (pour détails, voir
*                                 'iert')
imrt    dc.b    18,18,20,20     * offset des adresses des
*                                 registres IMRA et IMRB
*                                 (pour détails, voir
*                                 'iert')

iermt   dc.b    $df,$fe,$df,$ef * valeur des masques à
*                                 utiliser pour les
*                                 registres IERA et IERB
*                                 (voir note 2)
imrmt   equ     iermt           * note 2
iprmt   equ     iermt           * note 2
isrmt   equ     iermt           * note 2

tcrtab  dc.b    24,26,28,28     * offset permettant
*                                 d'accéder aux registres
*                                 de controle (TACR, TBCR
*                                 et TCDCR) des timers .
*                                 Notez que C et D sont
*                                 controlés par le meme
*                                 registre TCDCR
tcrmsk  dc.b    $0,$0,$8f,$f8   * masque des timers. Voir
*                                 note 3
tdrtab  dc.b    30,32,34,36     * offset permettant
*                                 d'accéder aux registres
*                                 de données des timers.
```

Là se terminent les routines 'setimer', 'mskreg' et 'getmsk'. Je pense que j'en ai assez dis pour cette fois-ci et qu'il vaut mieux vous reposer un petit peu, en attendant le prochain numéro qui parlera de la routine 'initint' qui permet d'initialiser les interruptions du MFP. A elle seule, cette routine demande beaucoup de gymnastique . Le programme d'application vous sera également présenté dans le prochain article.

Abtine NAJAND

Note 1

L'opération consiste à pointer sur l'un des registres IERA ou IERB et d'y masquer la donnée. Pour cela le programme, suivant le numéro du timer sélectionné, va déterminer un offset (ou une valeur à ajouter) qui permet d'obtenir l'adresse absolue du registre et d'y masquer la donnée. Pour cela on ajoute l'offset obtenu à l'adresse de base + 1 du MFP (l'adresse de base est en hexadécimal $fffffa00 ce qui donne une adresse de base + 1 égale à $fffffa01) et on tombe sur l'adresse du registre . Par exemple en regardant à la page 10 de cet article à l'étiquette 'iert' vous verrez la suite 6,6,8,8 qui correspondent, pour les deux premiers, à l'offset du meme registre IERA, et pour les deux autres, à l'offset du meme registre IERB. Dans ce cas là pour obtenir l'adresse absolue de IERA, le programme effectue l'addition suivante $fffffa01 + 6 = $fffffa07, celle de IERB étant, quant à elle $fffffa01 + 8 = $fffffa09.

Dans ce meme systeme de recherche, l'adresse absolue du registre IMRA sera obtenu en faisant $fffffa01 + 18 = $fffffa13 (ne pas oublier que les valeurs sans le signe '$' sont des valeurs décimales, ainsi ici 18 équivaut à $12, d'où le résultat obtenu).

Note 2

Comment déterminer les masques correspondant à chaque timer et à chacun des registres IERA, IERB, IPRA, IPRB, ISRA, ISRB, IMRA et IMRB ? Pour cela il faut examiner de près la structure de chacun des 8 registres ci-dessus qui sont associés à la fonction d'interruption du MFP (le cas du registre vecteur (VR) est différent et ne nous intéresse pas pour le moment).

La programmation des bits de masque se fait selon la structure ci-dessous (seuls sont indiqués les noms des timers auxquels correspondent les bits de masque, les autres bits , utilisés notamment pour les ports d'E/S et l'USART sont indiqués par un signe 'X'). Notez la position de chaque bit :

	B7	B6	B5	B4	B3	B2	B1	B0
IERA	X	X	A	X	X	X	X	B

(structure identique pour IPRA, ISRA et IMRA)

Ainsi pour masquer le registre de validation des interruptions correspondant au timer A , il faut faire un ET LOGIQUE (and) de la valeur de IERA avec la valeur binaire 11011111. Comme vous pouvez le constater, la position du bit 0 correspond à la position du bit correspondant au timer A dans le registre IERA. En hexadécimal ce masque s'écrit $df d'où sa présence devant les étiquettes 'iermt', 'imrmt', 'iprmt' et 'isrmt' en première position.

De la meme manière, si l'on veut masquer le registre de validation des interruptions correspondant au timer B, il faut faire un AND entre la valeur du registre (obtenue en faisant (a2) dans la deuxième ligne de 'mskreg') et la valeur binaire 11111110 ($fe en hexa), ce qui donc explique la présence de $fe en deuxième position devant les étiquettes 'iermt', 'imrmt', ...

On peut poursuivre la structure des registres en donnant celle de IERB, les autres étant la copie de IERA quand ils se terminent par la lettre A, et la copie de IERB, quand ils se terminent par la lettre B. Ainsi la structure de IMRA est identique à celle de IERA donnée ci-dessus et celle de IMRB à celle de IERB donnée ci-dessous :

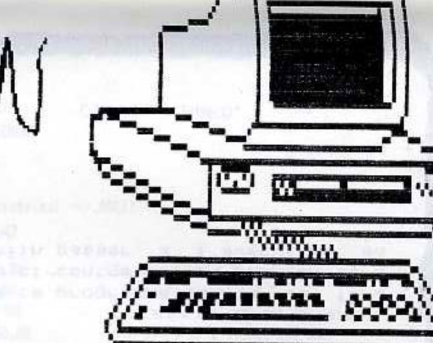
	B7	B6	B5	B4	B3	B2	B1	B0
IERB	X	X	C	D	X	X	X	X

(structure identique pour IPRB, ISRB et IMRB)

Le masque pour le timer D sera alors $ef, d'où la présence de cette valeur en quatrième position devant les étiquettes 'iermt', 'imrmt', 'iprmt' et 'isrmt'.

Note 3

La valeur du masque pour les registres de données des timers A à D est utilisée pour stopper le timer sélectionné. Dans le prochain article je vous donnerai le role précis de chaque bit de ces registres, en attendant je vous donne simplement les valeurs qui servent à stopper les timers . Dans l'ordre A, B, C, D ces valeurs sont $0,$0,$8f,$f8. Les explications détaillées vous seront données dans le prochain article.



PROGRAMMER EN ASSEMBLEUR



Je vous propose aujourd'hui de créer un premier programme en assembleur qui pourra tourner sous GEM ou TOS et dont le but est d'afficher sur l'écran les codes complets de chaque touche du clavier.

Pour atteindre ce but il nous faut disposer d'un EDITEUR, d'un ASSEMBLEUR ainsi que d'un EDITEUR DE LIENS (linker). Plusieurs versions existent sur le micro-ordinateur ATARI :

- la version CP/M 68K vendue par ATARI France avec le langage C aux développeurs pour environ 3500 Fr.

- la version Metacomco qui possède un éditeur pleine page très performant, un macro-assembleur et un linker.

- la version SEKA qui ne possède qu'un éditeur de ligne (bien mal pratique) et un assembleur n'acceptant pas les macro-instructions. Mais en contrepartie, il y a un débogueur bien utile pour la mise au point des programmes.

- la version Hisoft DevpacST qui a l'avantage de tourner sous GEM, avec un éditeur pleine page, un assembleur qui crée du code objet immédiatement exploitable (pas besoin de linker), et un désassembleur-déboggueur très performant.

Alors à vous de choisir...

DECODEUR DE CLAVIER

Le 68000 possède une instruction d'interruption logicielle (exception) nommée TRAP.

La syntaxe de cette instruction est :

TRAP #<numéro de vecteur>

Le numéro de vecteur étant codé sur 4 bits est donc un nombre compris entre 0 et 15. Nous disposons donc de 16 vecteurs différents permettant 16 interruptions logicielles distinctes. Ce numéro permet de trouver la nouvelle adresse devant etre chargée dans le compteur de programme.

Lorsqu'il rencontre cette instruction, le 68000 range la valeur du compteur de programme sur la pile utilisateur, ainsi que le registre d'état SR (Status Register). Le pointeur de pile SP (Stack Pointer = A7) est donc décrémenté de 6.

Puis le 68000 passe en mode superviseur en mettant le bit S du registre d'état à 1, et verrouille le mode Trace en faisant passer T à 0.

Enfin le 68000 charge le compteur de programme avec le vecteur dont le numéro lui a été passé avec l'instruction TRAP.

Ces vecteurs sont rangés en mémoire vive, dans la zone de données du superviseur (DS = Data Supervisor), aux adresses :

$80	vecteur n°0
$84	vecteur n°1
-	
-	
$BF	vecteur n°15

ATTENTION : Lors d'un appel de TRAP le 68000 ne sauvegarde pas sur la pile le contenu de ses registres et modifie en général le contenu de D0, D1 et A0. Il est donc conseillé de sauver ces trois registres sur la pile si le programme de l'utilisateur veut réutiliser leur contenu.

--*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

Il y a dans GEM 3 sections que l'on peut appeler à partir d'un programme utilisateur :

- GEM DOS qui est le DOS (Disk Opérating System) classique de tous les ordinnateurs, et que l'on appelle par TRAP #1.

- GEM VDI qui est le gestionnaire graphique de l'écran et de l'imprimante, que l'on appelle par TRAP #2.

- GEM AES qui est le gestionnaire de la souris et des diverses fenetres disponibles sur l'écran et que l'on appelle également par TRAP #2.

ASSEMBLEUR

Nous nous limiterons pour le moment à l'étude et à l'utilisation de GEMDOS. Grace à l'exception TRAP #1 nous allons pouvoir passer, via la pile, des paramètres au système.

Suivant la valeur de ces paramètres, le ST agira de différentes facons :

- si le mot (.W = word) placé sur la pile avant l'appel de l'exception TRAP #1 est un 1 ($0001), le ST ira lire le clavier et ramènera le code de la touche enfoncée dans D0 tout en écrivant le caractère correspondant sur l'écran (écho).

- si l'on y place au contraire le mot $8, le meme effet sera produit mais sans écho.

- Si l'on place successivement sur la pile le code ASCII (sous forme d'un mot de 16 bits) d'un caractère, puis le mot $2, on écrira ce caractère sur l'écran à la position courante du curseur.

- si au lieu du mot $2 on place après le code ASCII le mot $5, on écrira le caractère sur l'imprimante.

- si sur la pile on place le mot $0 avant l'appel de TRAP #1, on provoque une réinitialisation du système avec retour au bureau.

* Notre programme sera donc construit de la facon suivante :

1) - Initialisation de la pile et de la mémoire temporaire de sauvegarde.

2) - Lecture d'une touche au clavier avec écho sur l'écran et sauvegarde du résultat (D0) dans la mémoire temporaire.

3) - Si la touche enfoncée est un Esc (échappement) on revient au système.

4) - Sinon :

- on affiche un espace sur l'écran,

- on convertit le mot long contenu dans D0 en 8 caractères ASCII successifs permettant d'écrire la valeur hexadécimale de D0 sur l'écran,

- après avoir écrit le code de la touche on provoque un retour chariot (RC=$0D) suivi d'une descente d'une ligne (Line Feed, LF=$0A),

- puis on retourne au 2) .

* Comment écrire le contenu de D0 sur l'écran :

D0 est un long mot de 32 bits. Il s'écrit donc en hexadécimal à l'aide de 8 codes ASCII, chaque code représentant la valeur hexadécimale du 1/2 octet concerné.

Il faut commencer par écrire le 1/2 octet de poids le plus fort, puis le suivant, ...

On isolera chaque demi-octet à l'aide d'une instruction de décalage logique vers la droite (LSR). L'avantage de cette instruction est que simultanément au décalage vers la droite, elle entre des zéro par le bit de plus fort poids. Le nombre de décalage est indiqué dans l'opérande.

Exemple : D0 contient $F0520001

l'instruction LSR.L #28,D0

amènera le résultat----------D0 $0000000F

Ayant ainsi isolé ce 1/2 octet de plus fort poids, on peut chercher son code :

- si ce 1/2 octet est inférieur à 10, on obtient son code ASCII en lui ajoutant $30.

- si ce 1/2 octet est supérieur à 9 (lettres A, B, C, ...F) son code ASCII est obtenu en lui ajoutant $37.

Il suffit alors de ranger ce code ASCII sur la pile sous forme d'un mot de 16 bits, puis d'y mettre le mot $2 et d'appeler GEMDOS.

Pour accéder ensuite au 1/2 octet suivant, il faut faire LSR.L #24,D0 (D0 ayant été restauré avec sa valeur initiale).

Le glissement etant variable on utilisera l'instruction LSR sous sa forme dynamique :

LSR.L D1,D0

ou D1 sera initialisé à 28 puis décrémenté de 4 en 4.

On obtiendra alors le programme CODE.S qui une fois assemblé et listé fournira le fichier de listage ci-joint.

L'assemblage de ce programme source se fera comme indiqué sur l'écran ci-contre. Puis CODE.O sera linké et l'on pourra enfin passer à l'éxécution en cliquant deux fois CODE.PRG

Olivier HARD.

```
MC68000 ASSEMBLER VERSION 10.200

LOC      OBJECT              SOURCE STATEMENT

                      *
                      *    PROGRAMME DE DECODAGE DES TOUCHES DU CLAVIER DU 520 ST
                      *

0000'                        TEXT             Zone du texte source

                      *
                      *    --- Initialisation ---
                      *

                      *    Les initialisations se font a l'aide de l'instruction LEA
                      *    (Load Effective Adress) par rapport au compteur de programme.
                      *         L'avantage de cette methode est de rendre le programme
                      *    entierement translatable.

                      *                               Codes ASCII

         =001B             ESC     EQU     $1B          Escape
         =000D             RC      EQU     $0D          Retour Chariot
         =000A             LF      EQU     $0A          Line Feed


0000'    4FFA 087E         INIT    LEA.L   PILE(PC),A7  On initialise le pointeur de pile
0004'    47FA 087E                 LEA.L   MEMO(PC),A3  A3 pointe une memoire temporaire

                      *
                      *    --- Programme principal ---
                      *

0008'    616A              DEBUT   BSR.S   LECT         Lire une touche au clavier
000A'    2680                      MOVE.L  D0,(A3)      Ranger son code dans MEMO
000C'    0C00 001B                 CMP.B   #ESC,D0      Est-ce la touche ESC ?
0010'    670E                      BEQ.S   FIN          Si oui, alors FIN.
0012'    203C 0000 0020            MOVE.L  #' ',D0      On met le code de l'espace dans D0
0018'    6134                      BSR.S   OUT          Et on l'envoie sur l'ecran
001A'    610A                      BSR.S   AFFICHE      Puis on affiche le contenu de D0
001C'    6144                      BSR.S   LFCR         On envoie un RC et un LF
001E'    60E8                      BRA.S   DEBUT        Et on recommence.

0020'    4240              FIN     CLR.W   D0
0022'    3F00                      MOVE    D0,-(A7)     On met a zero le dessus de la pile
0024'    4E41                      TRAP    #1           Et on appelle GEMDOS.


                      *
                      *    --- Affichage d'une adresse hexadecimale ---
                      *

0026'    223C 0000 001C    AFFICHE MOVE.L  #28,D1       On prepare les 28 glissements

002C'    2013              AFF     MOVE.L  (A3),D0      On recupere le code dans MEMO
002E'    E2A8                      LSR.L   D1,D0        On glisse de D1 bits
0030'    6106                      BSR.S   CONVER       Et on convertit.
0032'    5941                      SUBQ    #4,D1        D1 vaut maintenant D1-4
0034'    6CF6                      BGE.S   AFF          S'il en reste on recommence,
0036'    4E75                      RTS                  Sinon on retourne.


                      *
```

```
MC68000 ASSEMBLER VERSION 10.200

LOC      OBJECT              SOURCE STATEMENT

                      *    --- Conversion hexadeimal --> ASCII ---
                      *

0038'    0280 0000 000F    CONVER  AND.L   #$F,D0       On masque les 4 bits faibles
003E'    0C00 0009                 CMP.B   #9,D0        Le nombre est-il > 9 ?
0042'    6E06                      BGT.S   LETTRE       Oui, alors son code est une lettre.
0044'    0600 0030                 ADD.B   #$30,D0      Non alors on n'ajoute que $30
0048'    6004                      BRA.S   OUT          pour avoir son code ASCII

004A'    0600 0037         LETTRE  ADD.B   #$37,D0      Pour les lettres il faut ajouter $37


                      *
                      *    --- Sortie caractere ---
                      *

004E'    48E7 FFFE         OUT     MOVEM.L D0-D7/A0-A6,-(A7)   Sauvegarde des registres

0052'    3F00                      MOVE    D0,-(A7)     On range le caractere sur la pile
0054'    3F3C 0002                 MOVE    #2,-(A7)     Puis le parametre ecran
0058'    4E41                      TRAP    #1           Et on appelle GEMDOS
005A'    588F                      ADDQ.L  #4,A7        Puis on restaure la pile.

005C'    4CDF 7FFF                 MOVEM.L (A7)+,D0-D7/A0-A6   On recupere les registres

0060'    4E75                      RTS                  Et on rentre

                      *
                      *    --- Retour chariot et descente ---
                      *

0062'    203C 0000 000D    LFCR    MOVE.L  #RC,D0       D0 contient le retour chariot
0068'    61E4                      BSR.S   OUT          On l'envoie sur l'ecran
006A'    203C 0000 000A            MOVE.L  #LF,D0       D0 contient le Line Feed
0070'    61DC                      BSR.S   OUT          On l'envoie sur l'ecran
0072'    4E75                      RTS                  Et on rentre.

                      *
                      *    --- Lecture d'une touche au clavier ---
                      *

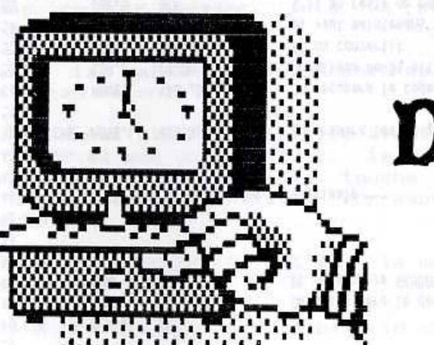
0074'    4280              LECT    CLR.L   D0           On efface D0
0076'    3F3C 0001                 MOVE    #1,-(A7)     On passe le parametre de lecture
007A'    4E41                      TRAP    #1           sur la pile, et on appelle GEMDOS
007C'    548F                      ADDQ.L  #2,A7        On restaure la pile
007E'    4E75                      RTS                  Et on rentre


0080'    =0800                     DS.L    512          Reservation de 512 octets
0880'    =0004             PILE    DS.L    1            sur la pile.

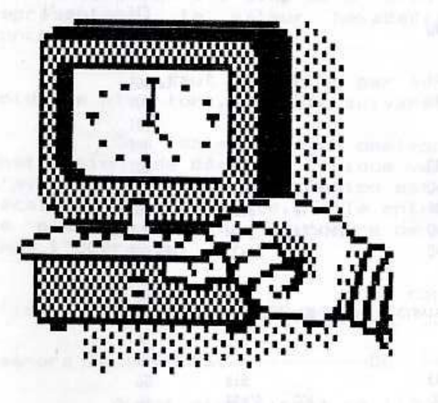
0884'    =0004             MEMO    DS.L    1            Memoire temporaire

                                   END

No errors found in this Assembly
```



MONTRE DIGITALE EN C



```
/*Ici on présente une montre digitale. Il s'agit d'un "DESK ACCESSOIRE"
I.e. un programme qui tourne en permanence et qui affiche l'heure
dans une petite fenêtre quelque part dans l'ecran.
Pour faire tourner ce programme chez vous il faut:
1. Compiler le source ci dessus.
2. Linker le code objet resultant de la compilation avec Link68 ou
 un autre programme d'édition de liens aussi nommé "Linker".
*/

/*
 * Desk Accessoire: Montre Digitale
 * Ecrit par Jan Gray, 1986.
 * Adaptation pour ST Magazine par Jakob Navia.
 *
 * link: accstart,osbind,montre,aesbind,osbind
 */

#include "define.h"
#include "gemdefs.h"
#include "osbind.h"

#define NO_WINDOW       -1              /* pas de fenetre ouverte */
#define NO_POSITION     -1              /* fenetre n'a pas de position */
extern long gemdos();
typedef struct window {
        int     id;
        int     x;
        int     y;
        int     w;
        int     h;
} Window;

/*
 * Il y a pas besoin de déclarer tellement de variables  globales!!!
 */

main()
{
        int     menuID;
        extern int      gl_apid;

        appl_init();
        menuID = menu_register(gl_apid,  "  Montre Digitale");
        events(menuID);
}

/*
 * Ici on voit le schéma general d'un desk accessoire: C est un programme
 * qui tourne en permanence et qui se reveille soit à un appel de GEM ou
 * avec le timer chaque 30 secondes. GEM communique avec le programme par
 * la variable event, le résultat du event_multi, la fonction multitaches
 * du GEM.
 * Le message de GEM est traité dans le switch (msgbuf[0])
 */
events(menuID)
int     menuID;
{
        Window  wind;
        int     event;
        int     msgbuf[8];
        int     ret;

        wind.id = NO_WINDOW;
        wind.x = NO_POSITION;

        for (;;) {
                event = evnt_multi(MU_MESAG | MU_TIMER,
                                0, 0, 0,
                                0, 0, 0, 0, 0,
                                0, 0, 0, 0, 0,
                                msgbuf, 30000, 0,
                                &ret, &ret, &ret, &ret, &ret, &ret);
```

ST MAG

C

```
	if (event & MU_MESAG) switch (msgbuf[0]) {
	case AC_OPEN:
		if (msgbuf[4] == menuID)
			if (wind.id == NO_WINDOW)
				openWindow(&wind);
			else
				wind_set(wind.id, WF_TOP, 0, 0, 0, 0);
		break;
	case AC_CLOSE:
		if (msgbuf[3] == menuID)
			wind.id = NO_WINDOW;
		break;
	case WM_CLOSE:
		if (msgbuf[3] == wind.id)
			closeWindow(&wind);
		break;
	case WM_MOVED:
		wind_set(wind.id, WF_CURRXYWH, msgbuf[4], msgbuf[5],
			msgbuf[6], msgbuf[7]);
		wind.x = msgbuf[4]; wind.y = msgbuf[5];
		wind.w = msgbuf[6]; wind.h = msgbuf[7];
		break;
	case WM_NEWTOP:
	case WM_TOPPED:
		if (msgbuf[3] == wind.id)
			wind_set(wind.id, WF_TOP, 0, 0, 0, 0);
		break;
	}
	if (event & MU_TIMER && wind.id != NO_WINDOW)
		update(&wind);
    }
}

/*
 * Extraire la date du Format DOS
 */
#define MINS(t)		((t >> 5) & 0x3F)
#define HRS(t)		(t >> 11)

#define DIGIT(d)	((d) + '0')

#define TEMPLATE	"hh:mm AM"
#define TEMP_LEN	8

update(wp)
Window	*wp;
{
	static char	time[]	= TEMPLATE;
	unsigned	t	= Tgettime();
	unsigned	hrs	= HRS(t);
	unsigned	hrs12	= (hrs % 12 == 0) ? 12 : hrs % 12;
	unsigned	mins	= MINS(t);

	/*
	 * sprintf n'est pas utilisable: elle consomme trop de memoire.
	 * Chaque accessoire doit consommer le moins possible de ressources
	 * et de memoire pour laisser de la place aux programmes que
	 * l'utilisateur est en train d'utiliser.
	 */
	time[0] = (hrs12 >= 10) ? DIGIT(1) : ' ';
	time[1] = DIGIT(hrs12 % 10);
	time[3] = DIGIT(mins / 10);
	time[4] = DIGIT(mins % 10);
	time[6] = (hrs < 12) ? 'A' : 'P';
	/*
	 * Ici on met l heure dans le titre de la fenetre avc wind_set
	 */
	wind_set(wp->id, WF_NAME, time, 0, 0);
}

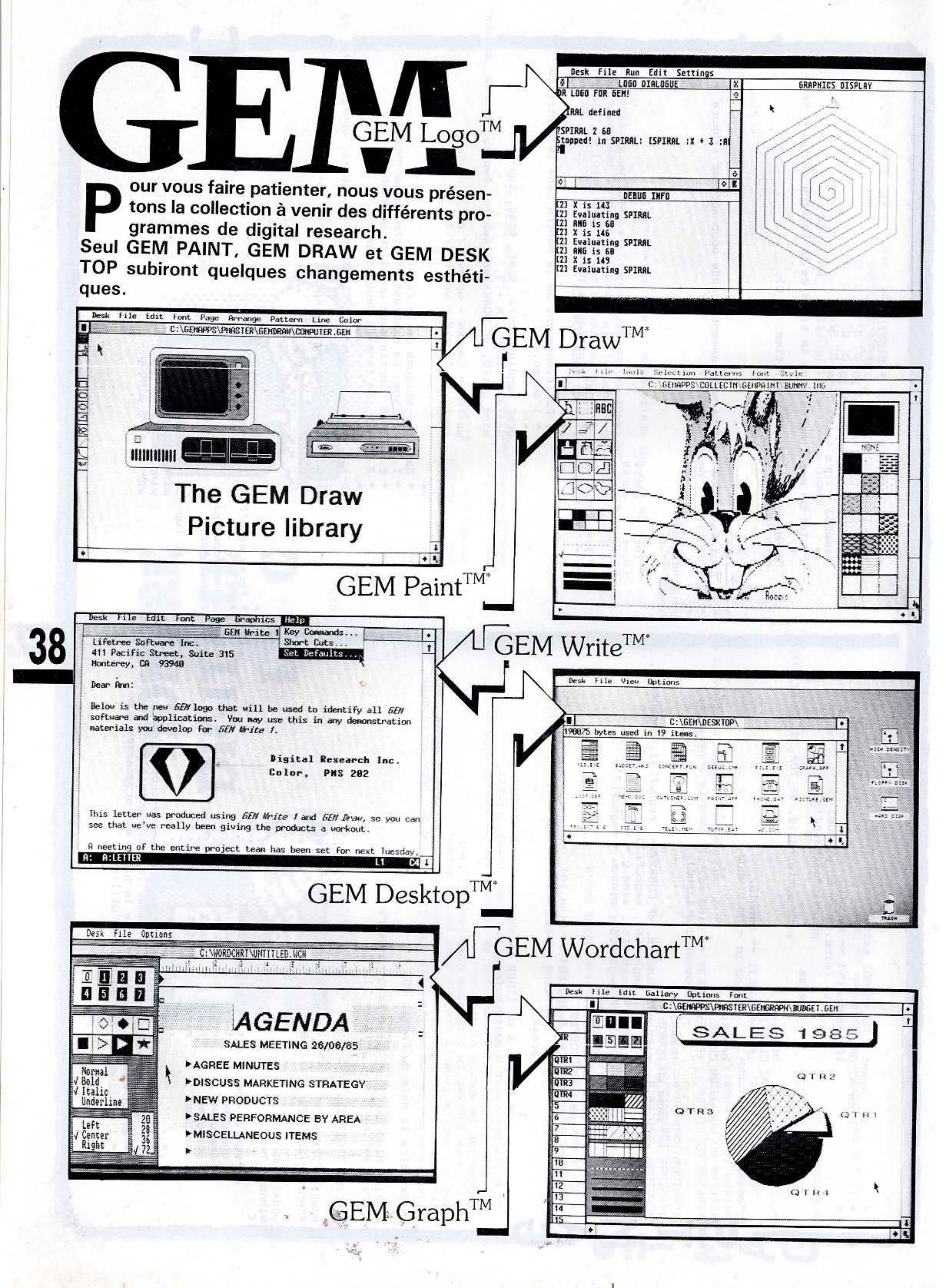
/*
 * Il faut faire une fenêtre sufisamment grande pour montrer l|heure
 * dans le titre ( de la fenêtre)
 */
openWindow(wp)
Window	*wp;
{
	int	workW;
	int	workH;
	int	ret;

	if (wp->id == NO_WINDOW) {
		if (wp->x == NO_POSITION) {
			/*
			 * Positionnement de la montre dans le centre de
			 * l'écran.
			 */
			graf_handle(&wp->w, &ret, &ret, &wp->h);
			wp->w *= TEMP_LEN + 3;
			wind_get(0, WF_WORKXYWH, &wp->x, &wp->y, &workW, &workH)
;
			wp->x += (workW - wp->w) / 2;
			wp->y += (workH - wp->h) / 2;
		}
		/*
		 * Caracteristiques de la fenetre:
		 * 1. NAME (Nom: La date est affiche comme "Nom")
		 * 2. CLOSER (Sortie avec un click a gauche)
		 * 3. MOVER (La fenêtre peut bouger)
		 */
		wp->id = wind_create(NAME|CLOSER|MOVER, wp->x, wp->y, wp->w, wp
->h);
		wind_open(wp->id, wp->x, wp->y, wp->w, wp->h);
		update(wp);
	}
}

/*
 * Ferme et efface la fenetre.
 */
closeWindow(wp)
Window	*wp;
{
	if (wp->id != NO_WINDOW) {
		wind_close(wp->id);
		wind_delete(wp->id);
		wp->id = NO_WINDOW;
	}
}
```

GEM

Pour vous faire patienter, nous vous présentons la collection à venir des différents programmes de digital research.
Seul GEM PAINT, GEM DRAW et GEM DESK TOP subiront quelques changements esthétiques.





EMULATEUR MS-DOS

EMULATEUR MS-DOS :
IBM PC/Atari ST
même combat

Dites-moi, mon bon monsieur, votre ST est ce qu'il compatible IBM ?
Cette question, si on ne me l'a pas posé 100 fois ?
Reprenons les choses au début. Des compatibles IBM, il y en a maintenant tant et plus. Des chers et des bon marché, des COMPACQ et des compacts, des portables et des qui faut être fou pour avoir envie de bouger.
Si le ST avait été un compatible de plus, on aurait pas pris la peine de faire un magazine spécial pour lui. Bâti autour d'un 68000, il ne peut pas être directement compatible avec des machines 8088 ou 8086. Cependant Atari était conscient dès le départ de l'importance de la famille IBM dans le monde de la micro informatique. Il a donc rendu sa machine le plus compatible possible avec IBM, tout en la dotant d'un processeur beaucoup plus puissant. Ce qui veut dire par exemple, qu'on peut connecter au ST les mêmes périphériques que sur PC. Les imprimantes compatibles PC sont compatibles ST. Ainsi que modems, tablettes à digitaliser...
Encore plus fort, le formatage des disques est le même. En connectant un lecteur 5'1/4 au ST, on peut donc relire des fichiers IBM. Des fichiers DBASE III ou LOTUS, par exemple.
Avec l'émulateur MSDOS/PC présenté par ATARI à Hanovre, on va pouvoir disposer sur le ST de la plus imposante bibliothèque logicielle existante et ceci avec deux avantages précieux : on possédera un matériel beaucoup plus puissant, qui pourra passer les cinq prochaines années sans rougir face à de nouveaux concurrents et aussi, et ce n'est pas le moindre des avantages, à un prix sans commune mesure avec l'IBM.

Cet émulateur est un émulateur **HARD**, ce qui veut dire que les applications tourneront aussi vite que sur l'original, contrairement aux machines qui ont un émulateur SOFT. (Suivez mon regard). La boite se connecte par l'interface disque dur. Elle contient un 8088 à 8 Mhz, 512K de mémoire et la place pour un coprocesseur arithmétique 8087. Le prix annonçé est d'environ 3500 francs et il faudra compter 1500 francs pour un lecteur 5'1/4.
Disponibilité Juin. A cet époque la compatibilité devrait être texte et graphique. A Hanovre seul des applications tout texte était présentées.

MEMSOFT/ATARI

MEMSOFT/ATARI :
LES LONGUES
FIANÇAILLES

On en parlait depuis Octobre, signera ? signera pas ?, les amours contrariés de Memsoft ont alimentés nos longues soirées d'hiver.
C'est désormais officiel, MEMDOS rebaptisé pour l'occasion MEMSOFT tournera sur le 1040, permettant ainsi la disponibilité immédiate d'un grand nombre de logiciels de gestion distribué par MEMSOFT, ainsi que du plus grand nombre encore d'applications verticales developpés par des éditeurs indépendants.
La configuration matériel minimum : une unité centrale 1 Mega, un disque dur 10 ou 20 mégas.
Devrait être disponible à partir de juin :
VENTES, un logiciel multi sociétés qui gére les stocks, les clients, la facturation. Paramètrable, il s'adapte aux spécificités de l'entreprise : TVA, représentants, remises spécifiques, bons de livraisons, en-têtes de facture, etc. .
Il interagit avec COMPTABILITE.
COMPTABILITE est destiné aux PME, professions libérales et artisans. Il permet l'édition instantanée de la balance et la tenue en temps réel de tous vos comptes. C'est aussi un outil de gestion permettant de suivre la marche de l'entreprise.
PAYE permet de maîtriser tous les éléments paye d'une entreprise : bulletins de paye. charges sociales...
Paramétrable aux spécifités de votre entreprise, le logiciel contient de nombreux écrans d'aide comme tous les logiciels MENSOFT.

PI-COMPTA

Les premiers logiciels de gestion pour ST sont maintenant disponibles. PI COMPTA est un logiciel de comptabilité éprouvé, qui fonctionne depuis quelque temps sous CP/M 80. Un émulateur CP/M étant disponible pour le ST, les utilisateurs de 520 ST ou 1040 peuvent disposer d'un programme ayant déjà tourner dans diverses entreprises, ce qui garantit un debuggage complet.
Très facile à utiliser, elle a une capacité de 350 comptes, 15 journaux généraux et 2500 écritures par période. Il y a 4 périodes par an. La saisie des écritures se fait sous un masque qui gère automatiquement la validité de l'introduction. Une calculatrice est disponible à tous moments. Un tableur est également fourni, il permet de reprendre des comptes pour une analyse financière.

ST-FACT

ST FACT a été developpé en Pascal OSS. Il fonctionne sur toute machine de la ligne ST, sous TOS et GEM. Il s'agit d'une facturation associée à une gestion de stocks et une gestion clients. Particulièrement facile à utiliser (souris,

LASERBASE

LASERBASE :
du Mac au Jack

Developpé sur le Mac, Laserbase sur ST combine la puissance d'une base de données avec une grande facilité d'utilisation. Ca c'est le créateur qui le dit, mais notre première approche semble bien le confirmer.
La création des fiches est entièrement graphique, en cliquant sur des symboles et en tirant des fenêtres avec la souris. Ce qui donne une fiche facile à lire, comme dans DB MASTER ONE. Mais là s'arrête la ressemblance. Laserbase ne charge pas les fiches dans la iémoire centrale mais les laisse dans le disque ou le disque dur. De ce fait le nombre de fiches n'est pas limité comme dans DB MASTER ONE et en fait le fichier peut aller jusqu'à 16 millions de caractères si vous avez un disque dur.
Les tris et les recherches peuvent se faire jusqu'à cent critères. Les impressions sont aussi faciles à créer que les fiches. Un regard sur ces performances vous permettra de juger si votre fichier est possible à créér avec Laserbase :

Nombre de champs : 1000
Taille maximum d'un champ 900 caractères
Taille maximum d'une fiche 30000 caractères
Taille maximum d'un fichier 16 Megaoctets

PI COMPTA et ST FACT

menus déroulants, boites de dialogue), il n'est pas protégé et fonctionne donc sans problème si un disque dur est requis par le grand nombre d'articles en stocks. La gestion clients permet un état des sommes dues ainsi qu'un cumul du chiffre réalisé avec chacun d'entre eux.
ST FACT est proposé à un prix très intéressant, tout à fait dans la philosophie du ST, puisqu'il est facturé à 995 francs Hors Taxe.

MAX MEGAMAX MEGAMAX MEGA

UN SYSTEME
DE DEVELOPPEMENT
COMPLET

Jusqu'à ces derniers temps, un programmeur sérieux ne pouvait se passer du système de développement proposé par ATARI. Le prix en était un peu élevé et la documentation fournie avec, présentait certains sous-ensembles (GEM, Kermit, ...) plus orienté IBM que ST.
Le nouveau prix d'une configuration un méga et l'apparition du sytème de developpement MEGAMAX vont mettre la création de programmes sur ST a la portée d'un encore plus grand nombre de programmeurs.
Le MEGAMAX C'est un système complet pour le ST. Il a servi a developper le jeu MEGAROIDS, un clone du célèbre jeu d'arcade ASTEROIDS, qu'un grand nombre d'entre vous doivent connaitre puisqu'il est désormais dans le domaine public.
(Une offre dans ce numéro vous permet de vous le procurer à bon compte)
Le MEGAMAX C comprend :

* Un compilateur C
Les conventions de Kernighan et Ritchie ont été suivie à la lettre afin de créer un langage standard facilement portable. Des fonctions absentes d'autres systèmes sont içi impléméntées : virgule flottante (32 et 64 bit), registre de variables, ...
La compilation se fait en une passe, elle génère un code objet directement. Il n'y a pas d'étapes intermédiaires d'assemblage. Le compilateur traduit du code à la vitesse de 1000 lignes par minute avec une unité de disquette et de 2500 lignes par minute avec le disque dur.
A tout moment dans votre code source C vous pouvez mélanger du code assembleur pour optimiser la vitesse d'execution.
MEGAMAX C supporte 6 registre de variables (2 d'adresses et 4 de données)
* Resource Construction Program
Ce programme permet la création immédiate de menus déroulants, de boites de dialogue, et la fabrication d'icônes. Ils sont créées graphiquement et générés sans écrire une ligne de code.
* Linker Sélectif
Il permet de linker seulement le code nécessaire à partir de la librairie. Cette sélectivité permet des tailles de code beaucoup plus petites que celles générés par des systèmes non sélectifs.
* Librarian
Le système comprend un « bibliothecaire » qui groupe les fichiers objets en un fichier homogène. Au linkage, vous donnez au linker le nom du fichier finissant par . lib et il cherche le code requis.
* Make
Make accomplit automatiquement les opérations nécessaires pour créer un nouveau fichier exécutable.
* Editeur
4 fichiers sont éditables simultanément. Fonctionne avec la souris.
* Code Improver
Il s'agit d'un processus optionnel ayant lieu en fin d'opérations. Il fournit un code qui est environ 10% plus court et 3% plus rapide.

* Accessoires de bureau
Il est possible d'écrire des accessoires de bureau en C qui pourront être utilisés en même temps que tout autre programme.
Megamax C supporte tous les appels à GEM et au BIOS Atari. Il est fourni avec une complète documentation sur GEM. Les routines AES. VDI. GEMDOS sont entièrement documentées. Des exemples de programmes sont fournis.
A noter que Megamax C n'est pas protégé et que les programmes développés n'ont pas a payer de royalties.

Pour ceux qui ont l'habitude du C DRI fourni dans le kit de développement ATARI, voici quelques comparatifs de temps de compilation, de taille mémoire et de temps d'exécution de quelques routines C classiques :

MEGAMAX C C DRI
Compil. / Exec. / Taille Compil. / Exec. / Taille
Sieve 1'34 2'28 6049 6'07 2'47 11852
Fibonacci 1'30 17'6 5975 5'59 16'4 11746
Intmath 1'32 3'73 6411 6'27 3'72 12180

DIGITALISATION

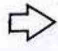
DIGITALISATION VIDEO :
le ST ouvre les yeux !

Avril marque le début de la digitalisation vidéo sur ST. On attendait MICRO COMPACT, une société française qui avait annonçé ce produit il y a plusieurs mois, ce sont PRINT TECHNIK (RFA) et HIPPOPOTAMUS (USA) qui arrivent les premiers.
PRINT TECHNIK présente VIDEO DIGITIZER qui digitalise n'importe quel signal vidéo (NTSC, PAL, SECAM) venant de toutes sources (Caméra, magnétoscope, TV). Il faut cepandant que l'écran de digitalisation reste fixe pendant 5 secondes, ce qui est le laps de temps mis par le digitaliseur a balayer l'écran. Deux versions du logiciel sont disponible, une pour l'écran couleur avec une résolution de 320 sur 200 en 16 nuances de gris. L'autre pour l'écran monochrome en 620 sur 400 mais avec seulement du noir et du blanc. Les images peuvent être sauvées sur disquette en format standard ou en format DOODLE (Version monochrome) ou en format Neochrome (Version couleur). Elles peuvent être également mis en diaporama pour des démonstrations publicitaires.
Une table des couleurs permet également d'assigner une quelconque des 512 nuances à chaque niveau de gris. Les images peuvent être réorientés , mises en négatif, zoomées, et enfin imprimés sur imprimante monochrome (type EPSON) ou couleur (Canon PJ 1080).

Celui d'HIPPOPOTAMUS semble encore plus puissant, mais nous ne l'avons pas encore vu dans sa version définitive. En version monochrome, il digitalise en 1/60 de seconde et en version couleur en 1/2 seconde. On peut donc faire du dessin animé (3 secondes sur un 1040) à 30 images par seconde. Les images peuvent être retravaillées avec Neochrome ou Degas.

TRANSYLVANIA AND THE CRIMSON CROWN

Ils faisaient partie des premiers jeux d'aventures sur l'ATARI 520 ST, ces deux logiciels étant des transfuges de l'apple II. Dans TRANSYLVANIA, vous êtes chargé de libérer la princesse Sabrina qui a été enlevée par un vampire, et ce, avant l'aube sous peine de la perdre. THE CRIMSON CROWN est la suite de cette histoire en sachant que vous n'êtes plus seul dans votre quête : après avoir libéré Sabrina, vous devez restituer à son père ses armes et sa couronne.
Ces logiciels sont corrects et ils plairont à tous. Il est seulement regrettable que le graphisme n'ait pas été un peu plus approfondi.
GENRE : AVENTURE.
INTERET : ***
GRAPHISME : **
BRUITAGE : /
RESOLUTION : COULEUR





DB MASTER ONE

Db master one se définit comme un gestionnaire de fiches à la fois simple et puissant. Son utilisation sous **GEM** avec menus déroulants, boîtes de dialogues et icones le rend rapide à mettre en œuvre. Sa puissance se détermine par la rapidité des tris effectués lors de la recherche de fiches sur les critères entrés par l'utilisateur. Le logiciel se divise en deux programmes sur la disquette : **makeone.prg** et **useone.prg**. Il est important de sauver fréquemment son travail, le logiciel ne possédant pas de sauvegarde automatique. Nous allons maintenant détailler l'utilisation des deux programmes.

MAKEONE.PRG :

Il sert a créer une fiche type, c'est-à-dire le masque de saisie que vous utiliserez pour remplir votre fichier. Il est à noter l'existence de fiches-types prêtes à l'emploi dans le menu **splat** (ces « instantanées » peuvent changer suivant les versions futures, par exemple le mailing list qui crée le format postal est en version américaine). Si vous voulez créez vous même votre fiche-type, il suffit de cliquer à l'endroit que vous voulez sur l'écran pour déclarer un champ d'information dans lequel vous entrerez le titre (nom, date, livre, prix d'achat, etc) puis déterminerez la taille de la zone de données grâce à la souris, sur des poignées de modification à l'extrémité droite de la zone de champ. A gauche, la poignée sert à déplacer votre zone de champ n'importe où sur l'écran, ce qui veut dire que vous pourrez réarranger l'esthétique de votre fiche-type, même d'un fichier déjà créé et utiliser grâce à « **redesign** » du menu file (une sorte de remise à neuf avec la possibilité de rajouter ou supprimer des champs sans modifier les autres données du fichier). Dans le menu **options** vous pourrez choisir la couleur, la taille, le style (épais, italique, souligné) pour la zone titre comme pour la zone de données, ce qui permet de mettre en valeur certaines parties de la fiche et de lui donner une esthétique hors-pair. Les fonctions **couper-coller** existent aussi pour l'édition. Il ne vous reste plus qu'à sauver votre fiche-type ce qui aura pour fonction de créer un fichier prêt à remplir grâce à useone.

USEONE.PRG :

Ce programme sert donc à remplir les fiches de votre fichier créé avec **makeone.** Cliquez le nom du fichier dans la zone sélecteur d'objet puis vous pouvez commencez à modifier des fiches déjà existantes ou en rajouter d'autres, le curseur se déplaçant uniquement dans les zones de données et en change avec un simple return (celui du clavier alphabétique, celui du numérique servant à changer de ligne dans une zone de données).

Mais la fonction essentielle de **Db master one** est de trier les fiches. Suivant la façon que vous désirez les voir s'afficher à l'écran, sur l'imprimante ou sur la disquette il vous suffit de créer un état de tri et de lui donner un nom (10 états peuvent être créés par fichier et sont sauvegardés avec lui). Pour créer un état de tri cliquez sur **edit**, puis choisissez la forme de votre état (en ligne, en colonne, tel quel sur l'écran pour pouvoir l'utiliser avec un traitement de texte et enfin sous forme d'étiquette au format postal). Vous devrez choisir ensuite les champs qui seront affichés lorsque le tri sera terminé suivant la forme choisie. Enfin le tri **(find).** Vous avez trois niveaux de tri, chacun pouvant prendre trois sortes de recherche : une par équivalence sur les premiers caractères ou chiffres du champ de données, une par encadrement et la dernière par la contenance d'une chaîne (caractères ou chiffres) dans toute la zone de données du champ considéré. Vous pourrez définir le sens de classement du tri, par ordre alphabétique ou inverse, l'affichage des sous-totaux et le changement de page si les données changent, tout ceci étant paramétrable pour chaque niveau de tri.

Dans le menu record, **« fin some »** vous permettra d'utiliser les mêmes fonctions de tri que **edit,** mais pour une visualisation directe des fiches rechercher en vue d'une remise à jour, dans le cas où votre fichier est trop important pour permettre une recherche en se déplaçant fiche par fiche **(next et previous). Add** permet d'ajouter une fiche vierge.

Conclusion :

Db master charge tout le fichier en mémoire, ce qui permet sa rapidité de traitement dans les recherches, mais le limite à 320 k caractères. Vous pouvez créer des fiches contenant 3.000 caractères répartis sur un maximum de 100 champs d'informations. **Db master** fonctionne dans les trois résolutions, mais un fichier créé dans l'une ne fonctionnera pas dans l'autre. Enfin, si votre fichier est très important et que votre recherche nécessite plus de trois tris alors il vous suffit de sauver un état de tri, de le recharger, puis de refaire un tri, ceci jusqu'à obtention du résultat voulu. Malgré ces limitations, il n'en reste pas moins que **Db master** couvrira largement les besoins de l'utilisateur courant et restera performant à l'usage. De par sa souplesse d'utilisation il séduira plus d'un utilisateur qui comprendra les possibilités offertes par un tel traitement informatique face à des fiches de carton qui limitent de beaucoup les possibilités de recherches et de classements de l'information.

Essai réalisé par :
Frédéric MARTON

BRATACCAS



Voici donc le premier jeu d'arcade utilisant pleinement les capacités d'animation graphique du 520 ST : *BRATACCAS*.

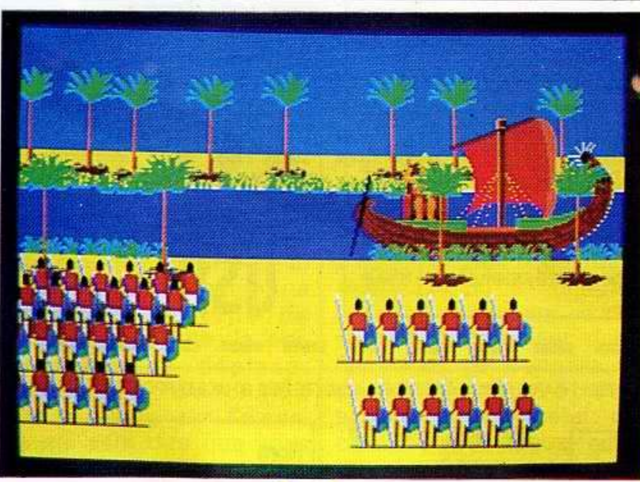
Vous incarnez un savant KYNE ayant mit au point une nouvelle race de surhommes. Mais la pègre et le gouvernement voulant s'en emparer, vous voilà contraint d'aller sur le satellite BRATACCAS pour prouver votre innocence.

Ce jeu est remarquablement bien conçu et l'animation est fantastique : les personnages semblent avoir une vie indépendante et un véritable dialogue s'instaure entre vous et eux. Vous pouvez y jouer sur un moniteur monochrome ou couleur ce qui rend ce logiciel accessible à tous.

Les mouvements du personnage principal sont très complets (courir, marcher, prendre des objets, pivoter à droite, à gauche, etc...), et peut s'effectuer indifféremment au clavier, au joystick ou à la souris.

En conclusion, ce logiciel préfigurant une nouvelle génération de jeux aurait été presque parfait si le maniement du joystick ne tenait pas du miracle.

GENRE : ARCADE.
INTERET : *****
GRAPHISME : *****
BRUITAGE : ***
RESOLUTION : COULEUR OU MONOCHROME.



St Art

KING QUEST II

KING QUEST fut un des premiers jeu d'aventure à être proposé sur la gamme ST, mais il y a cependant une particularité : le jeu est **animé**, c'est à dire que le maniement du personnage principal s'effectue en temps réel par l'intermédiaire du joystick (ou du clavier). Toutefois, il est possible de dialoguer avec l'ordinateur comme tout bon jeu d'aventure qui se respecte.

Vous voici revenu au Moyen-Age, et vous incarnez un roi qui est aimé et respecté de tous. Mais vous ne pensez qu'à une seule chose : delivrer une princesse qui est retenue prisonnière dans une tour et ce, malgré tous les pièges qui vous sont tendus au cour de votre longue quète, qui ira « de la maison de mère-grand », jusqu'aux fonds sous-marin.

Ce jeu est difficile et conviendra parfaitement à tous les amateurs d'aventures ; l'animation est réussie, et les couleurs sont franches. Au chapitre des regrets, il est dommage que la transposition du jeu venant de l'apple II ne profite pas pleinement des capacités du ST, le graphisme nous laissant un peu sur notre faim.

GENRE : AVENTURE ANIMEE.
INTERET : ****
GRAPHISME : ***
BRUITAGE : **
RESOLUTION : MONOCHROME OU COULEUR.

MEGAROÏDS

C'est une réplique du célèbre ASTEROÏDS qui fonctionne en couleur et en monochrome. La finesse du graphisme est très proche du graphisme « vecteur » du jeu d'arcade et l'effet relief est très agréable. Des soucoupes meurtrières viennent encore augmenter la difficulté déjà grande, au fur et à mesure des tableaux rendant ce jeu plus qu'agaçant, d'où son intérêt. Demandez-le à votre renvendeur, il est dans le domaine public, donc gratuit (disquette non comprise, bien sûr !). On regrettera seulement l'absence de l'option joystick, les commandes se faisant au clavier.

GENRE : ARCADE
INTERET : ***
GRAPHISME : ***
BRUITAGE : *
RESOLUTION : COULEUR/MONOCHROME.

MONKEY BUSINESS

C'est une réplique exacte de Donkey Kong avec d'autres personnages. Le décor est fin et le jeu rapide, mais l'animation n'est pas fameuse et il ne fonctionne qu'en monochrome.

GENRE : ARCADE
INTERET : **
GRAPHISME : *
BRUITAGE : *
RESOLUTION : MONOCHROME

DELTA PATROL

On ne le confond pas longtemps avec SUPER COBRA ou avec SCRAMBLE le seul but du jeu étant de détruire le plus d'ennemis possible sans tomber en panne de fuel ; il laisse une impression d'inachevé. Le jeu est rapide et l'animation de l'hélicoptère bonne mais le scénario est vraiment nul.

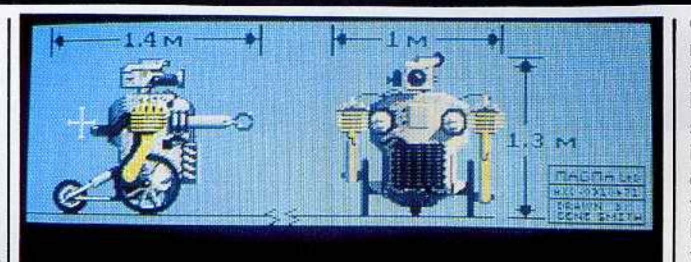
GENRE : ARCADE
INTERET : —
GRAPHISME : *
BRUITAGE : *
RESOLUTION : MONOCHROME

HACKER

MUD PIES

Ce jeu rappelle le jeu d'arcade FOOD FIGHT d'ATARI. Le but du jeu est de ramasser des gâteaux disséminés dans l'écran et tuer à l'aide de ceux-ci les clowns qui vous empêchent de sortir du terrain de jeux pour accéder aux tableaux supérieurs. Diverses nourritures permettent au joueur de survivre mais gare à l'indigestion.
C'est le 1er jeu d'arcade en couleur sur ST, musical (musique de cirque) ayant un bon graphisme et une bonne animation (mais nous nous verrons beaucoup mieux !) avec écrans bonus. Il ne fonctionne pas en haute résolution, manque d'intermissions et est trop facile (même dans les niveaux élevés !)

GENRE : ARCADE
INTERET : **
GRAPHISME : **
BRUITAGE : **
RESOLUTION : Couleur.



La célèbre firme Américaine ACTIVISION commence à adapter ses logiciels sur l'**atari ST**, et, fidèle à leur réputation, ils sont généralement toujours réussis.
HACKER n'échappe pas à la règle, car l'adaptation du célèbre jeu est une incontestable réussite : rappelons que pour ceux qui n'ont pas connu **HACKER** dans la version pour l'ATARI **130 XE** ou du commodore 64, il s'agit, après avoir réussi à entrer dans un système informatique, de parcourir les grandes capitales du monde pour récupérer les différents morceaux d'un document ultra-secret. Vous aurez à le négocier avec des espions, soit en achetant une partie de celui-ci, soit en l'échangeant avec un objet acheté auparavant. Mais un problème surgira lorsque ceux-ci s'obstineront à convoiter chacun le même objet en échange de leur précieuse parcelle de document. A vous d'essayer de leur donner autre chose...
Autre paramètre primordial, le jeu se déroule en temps réel : ne trainez donc pas en route, le tracé précis du souterrain vous sera indispensable sous peine de vous voir déconnecté du réseau pour avoir passé trop de temps à négocier une parcelle du document ou s'être stupidement perdu dans une impasse. D'autre part, au bout d'un certain temps, le système ayant repéré votre intrusion, des satellites de détections seront lachés à vos trousses, et si par malheur ils vous découvrent, ils vous réclameront à quatre reprises une clef d'accès à un niveau de sécurité supérieur, la première étant « MAGMA, LTD » ; quant aux autres, observez donc le début du jeu lorsque l'ordinateur vous demandera de répondre aux diverses questions concernant les robots.
D'une manière générale, le programme est remarquablement bien fini, le graphisme est très réussi, de même que l'animation. On attend la suite des logiciels ACTIVISION avec impatience. Un dernier conseil, n'oubliez pas de brancher votre imprimante, pour les honneurs en cas de succès...
GENRE : ARCADE.
INTERET : *****
GRAPHISME : ****
BRUITAGE : *
RESOLUTION : COULEUR OU MONOCHROME.

FIRST WORD

First word est enfin le traitement de texte convenable que l'on était en droit d'attendre sur une machine de la série ST. Il utilise pleinement les possibilités de l'atari sous Gem et vous fera rapidement laisser tomber tout autre traitement de texte fastidieux ou il faut retenir plusieurs dizaines de combinaisons de touches pour obtenir un document correct. Nous n'expliquerons pas ici l'utilité d'un tel logiciel (voir ST magazine n° 1 p. 34), nous allons plutôt décrire ses qualités.

Le manuel en anglais étant sur la disquette il vous sera facile de le lister sur imprimante, mais vous ne l'utiliserez quasiment pas ; le logiciel étant d'une telle facilité d'emploi grâce aux menus déroulants, dont un **Help**, qui vous précise l'utilisation de la fonction que vous désirez.

L'organisation de **First word** à l'écran est un modèle du genre, en effet, le texte que vous entrerez sera dans une fenêtre, donc paramétrable dans ses dimensions et sa position. Ce qui est très utile, puisque vous pouvez éditer quatre textes en même temps, chacun dans une fenêtre différente. Avec les fonctions couper-coller vous pourez transférer des zones de texte d'une fenêtre à l'autre (un exemple d'utilisation des quatre fenêtres : une pour l'écriture même de la lettre, une autre contenant des entêtes déjà préparées, une autre pour les formules de politesse de fin de lettres et la quatrième pour l'ajout d'un rapport écrit ultérieurement qui viendrait confirmer votre lettre).

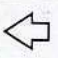
En bas de l'écran se trouve les icones des dix touches de fonctions (afin de pouvoir contrôler leur état), que vous pouvez mettre en œuvre avec la souris. Sur la moitié droite de votre écran se trouve une table contenant tous les caractères et symboles auxquels l'on ne peut accéder par le clavier (les accents courants du clavier français sont accessibles par celui-ci normalement). Elle est vraiment utile surtout pour des documents mathématiques contenant des lettres grecques et des symboles tels que l'intégral, les racines et bien d'autres.

First word ne possède pas plusieurs polices de caractères mais plusieurs styles, tel le souligné et l'italique. Il possède en outre le style exponentiation et indexation (**super** et **subscript**).

Toutes les fonctions de bases d'un bon traitements de texte y sont : couper-coller, recherche d'une chaîne et son remplacement par une autre, justification des paragraphes, reformatage de ceux-ci, centrage, réglage des indentations et bien d'autres.

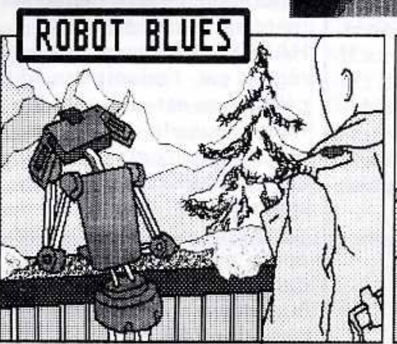
Pour ce qui est de bien voir sur l'imprimante ce que vous avez créer à l'écran, rien de plus simple : il y a un programme sur la disquette qui permet de configurer les sorties en fonctions de l'imprimante, le logiciel possédant plusieurs drivers. De plus pour imprimer des documents correctement, il suffit lors de la création d'un texte de choisir parmi trois options de sauts de pages, en déclarant dans la marge l'endroit ou l'on veut qu'elles s'effectuent (un saut inconditionnel et deux conditionnels : un au nombre de lignes par pages et l'autre pour ne pas casser des paragraphes). Avec **Layout** on peut créer des en-têtes et des pieds de pages, sur le centre, la droite et la gauche, ainsi que spécifier le nombre maximum de lignes par page et déclarer les marges horizontales et verticales, tout ceci très simplement avec des flèches que l'on clique dans des boîtes de dialogues.

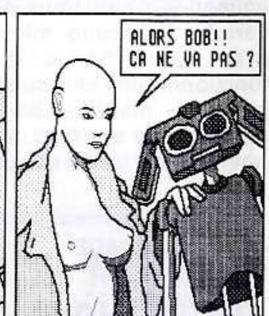
De par l'excellence de ses prestations, à l'usage, Fist word se rendra vraiment très vite indispensable, quel que soit le document à écrire et sa taille. De plus les prochaines versions de GST software permettront d'inclure des graphiques avec les textes. Le seul point noir est le manque de plusieurs polices de caractères et le choix de leurs tailles.



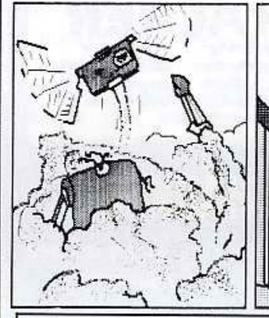
Essai réalisé par :
Frédéric MARTON

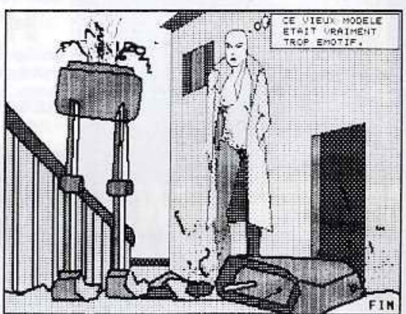












VOUS AVEZ DES CHOSES INTERESSANTES A DIRE SUR LE 520 ST !
VOUS ALLEZ OU VOUS AVEZ DEJA DEVELOPPE DES CHOSES SUR LE 520 ST !
VOUS AVEZ REMARQUE DES ERREURS OU DES MANQUES DANS LES ARTICLES DE ST MAGAZINE !
CONTACTEZ NOUS A :
COLLABORATION ST/PRESSIMAGE 210, RUE DU FAUBOURG ST-MARTIN, 75010 PARIS.

Cette bande dessinée a été réalisée avec le programme DEGAS, en haute résolution monochrome 640 × 400. Vous pouvez en réaliser vous-mêmes ; ST MAGAZINE publiera les meilleures d'entre-elles ; chaque publication rapportera à son auteur un abonnement d'un an au journal. Le thème est libre avec une préférence cependant pour l'érotico-fantastique.